

Project Thesis/Report

Contents

1	Introduction	4
2	Discrete-time Dynamical Systems	7
2.1	Invariant Sets	8
2.2	Chaos	10
2.3	Conjugacy	14
3	The Learning Problem	17
3.1	Taken Embedding Theorem	18
3.2	Practical Issues of Takens theorem	20
4	Driven Dynamical Systems to Forecast Problems	22
4.1	Nonautonomous and Driven Dynamical Systems	22
4.2	Unique Solution Property	25
4.3	Choosing the driven system g	29
4.4	The next step in Dynamics	31

<i>CONTENTS</i>	3
4.5 A discussion of G_T	33
4.6 Advantages of learning Γ	34
5 Implementation	37
5.1 Implementing G_T	37
5.2 Changes in this thesis/project/report to previous work of [?]	38
5.3 Numerical simulations	39
5.3.1 Double Pendulum	39
5.3.2 Other attractors I've experimented with: Clifford & Thomas	39
5.4 Generalized Pearson correlation coefficient for the DP and other attractors	40
6 Conclusion	42

Chapter 1

Introduction

Experimenting on biological, physical, and artificial systems in order to generate a more informative dynamical model is a well-established practice in modern science. Traditional methods for modelling physical systems are based on laws of physics that are based either on empirical relationships or on intuition. For systems that evolve with time, physical laws yield mathematical equations that govern how the quantities evolve with time. However our world around is, for the most part, much more complex than that which can be distilled into elegant equations. We do not fully understand many of the more complex systems, nor do they even provide us with a good physical intuition of the underlying principles governing the dynamics. To complicate this - the underlying systems we observe often display sensitive dependence on initial conditions; despite having highly similar initial conditions, differing orbits diverge quickly and to such an extent that it becomes seemingly impossible to retrace their steps back to the original conditions. Even the presence of usually 'negligible' computational noise or measurement error renders long-term point-wise prediction infeasible. There are many difficulties that one encounters while modelling such systems:

- i. One may not have access to the complete states of the systems
- ii. The actual system may be described by functions that behave wildly in the sense their graphs have a wild oscillatory behaviour, i.e., they have a large functional complexity.

Models derived primarily from data can be classified into three categories:

- i. Interpretable models (i.e., they establish relationships between internal physical mechanisms),
- ii. Partially interpretable models capturing some modes of the dynamics,
- iii. Non-Interpretable models, defined as such mainly due to the fact that they are defined on a different phase space that is usually high-dimensional.

Examples in the literature attempting to forecast data from such systems have tried a number of different approaches with differing degrees of success. When we have complete access to states of the

system, an ordinary differential equation can be obtained from data (Citations to Sindy and [?, ?]) wherein one could approximate the vector field by a library of functions to obtain interpretable models.

Recent partially interpretable models available in the literature have been based on the Koopman operator (e.g., [?, ?]) to employ observables mapping the data onto a higher-dimensional space. This then makes the dynamics in the higher-dimensional space more amenable for approximation by a linear transformation. Such methods, obviously, do not guarantee an exact reconstructions for nonlinear models, and in practice provide poor long-term consistency for a large class of systems. (Add citation)

The non-interpretable models include the delay embedding and the machine learning algorithms. One could learn a system conjugate to the underlying system by applying the Takens delay embedding [?] when one has “good” observations from the system; this learnt system could then be used for forecast the observed data. Takens delay embedding theorem [?] and its various generalisations (e.g., [?, ?, ?, ?]) establishes the learnability of a system constructed by concatenating sufficiently large previous time-series observations of a dynamical system into a vector (called delay coordinates). This then establishes the existence of a map on the space of delay coordinates equivalent (or topologically conjugate) to the underlying map from which the observed time-series was first obtained. Although topological conjugacy guarantees an alternate representation of the underlying system, the quality of this representation still depends on numerous parameters, making the comprehension of the dynamics rather unreliable (e.g., [?]). One reason for this fragility is that the embedded attractor in the reconstruction space not always an attractor of the map that is learnt in the reconstruction space, despite unquestionably being an invariant set. When the embedded object is not (**definitely/always?**) an attractor (as explained in Chapter 3) it can cause predictions to fail.

Practically, the application of Takens embedding involves learning a map through some technique and consequently one wishes that these would have low functional complexity[?], i.e., functions with fewer oscillatory graphs. Pure machine learning methodology that processes temporal information (like the echo state networks (Citations to echo state networks)) by mapping data onto a higher dimensional space for further processing. Although they perform well on forecasting some dynamical systems, they fail completely on others (Citations here) as there is often no guarantee that the right function was learnt during training.

This thesis deals with the implementation and analysis of non-interpretable models that can guarantee **exact/accurate** reconstruction. The project work concerns the study and implementation of a method (Citation to paper with Adriaan) that incorporates learning a function by mapping the data on to a higher dimensional space using what is called a driven dynamical system (See Chapter 4). With a clear understanding of how the data is mapped onto the higher dimensional space, the method then permits the learning of a dynamical system topologically conjugate to that of the underlying system. Instead of linear regression as in the training of echo state networks, deep learning methods are employed to learn the correct function. With slight modifications to

the implementation in the paper (with Adriaan), we show that one can construct accurate non-interpretable models with the ability to reconstruct attractors from more hard-to-forecast systems like the double pendulum. (The forecasting of the time-series from a double pendulum has not been reported before.) Moreover, we also demonstrate that long-term statistical consistency is preserved.

By solidifying the mathematical underpinnings of our theory, we hope to guarantee the ability to construct models with predictive power ranging from molecular biology to neuroscience. in the near future. (We can modify this sentence when the report is completed.)

The report is organised as follows:

In Chapter 2, we recall the definition of a discrete-time dynamical system, how a discrete-system arises from a flow of an ODE and then proceed to define the inverse limit space and topological conjugacy of autonomous systems.

In Chapter 3, we introduce the problem of forecasting dynamical systems, state the Takens delay embedding theorem and discuss various issues faced while forecasting.

In Section 4, a driven dynamical system is defined and discuss the properties of a specific class of driven dynamical systems that we make use of in this thesis/report/paper.

Finally in Section 5 we show the implementation of these forecasting methods, and conclusions are provided in Section 6.

Chapter 2

Discrete-time Dynamical Systems

In this chapter, we provide a brief description of what a discrete-time dynamical system is and what it means for it to exhibit chaos. We refer to [?, ?] for more details.

At its most elementary level, a dynamical system is just something that evolves deterministically through time. In the context of this thesis, deterministic refers to the fact that a system evolves according to specified rules rather than based on random events. Dynamical systems arise in a variety of situations. A continuous-time dynamical system describes the states for all values of the time. Specifically, if the motion of a pendulum in which the quantities such as the angular position and angular momentum are known at all times, then it is a continuous-time dynamical system. The equations of the dynamical system can take the form of one or more ordinary differential equations that determine the relevant quantities at any future time if we know the initial location and momentum. In ecology, discrete-time dynamical systems are widely used to model population growth. The model in this case is a function calculating the following generation's population given the population of the previous generation. If we know the starting population, we may once again calculate the population at any time in the future.

Formally, a function $T : U \rightarrow U$, where U is some set is a *discrete-time dynamical system* and its iterates $\{u, Tu, T^2u, \dots\}$, where T^n denotes the n -fold composition of T with itself, describe the evolution of an initial condition $u \in U$ (Note that we frequently drop the brackets and denote $T(u)$ by Tu so as to simplify notations).

Continuous-time dynamical systems modelled using ordinary differential equations can give rise to discrete-time dynamical systems. To see this, consider a differential equation $\dot{x} = f(x)$, $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $n \in \mathbb{N}$ given to have a unique solution passing through each point $x \in \mathbb{R}^n$, call it $x(t)$ where $x(0) = x_0$.

Definition 1 (Flow of an Equation). *The flow of the equation $\dot{x} = f(x)$ is defined to be a mapping $\varphi : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ where $\varphi(x_0, t) = x(t)$ for the solution $x(t)$ with $x(0) = x_0$*

By fixing $t = K \in (0, \infty)$, we can define the *time- K map* as $T(x) := \varphi(x_0, K)$, and it is easily verified that $T \circ T(x_0) = \varphi(x_0, 2K)$. In general the m^{th} iterate of x_0 under T would be the value of the solution of the ODE evaluated at time mK with the initial condition x_0 , i.e. $\varphi(x_0, mK)$. Thus ordinary differential equations give rise to a discrete-time dynamical system by sampling the value of the solution $x(t)$ at time intervals K units apart.

A numerical discretization of a differential equation can also give rise to a discrete-time dynamical system. For instance, Euler's method approximates $\dot{x}(t)$ by $(x(t+h) - x(t))/h$; if h is fixed throughout, the solution of a differential equation $\dot{x} = f(x)$ can be approximated at the time instant $t + (m+1)h$ by iterating the equation

$$x(t + (m+1)h) = x(t + mh) + hf(x(t + mh)) \quad (2.1)$$

Adopting more succinct notation by replacing $x(t + mh)$ with u_m , we rewrite the above equation as

$$u_{m+1} = u_m + hf(u_m) \quad (2.2)$$

or in even simpler terms as the discrete-time dynamical system with map $T(u) = u + hf(u)$, where T , u and f are understood to be as above.

Of course, discrete-time dynamical systems need not always arise through a differential equation, and once again we may consider the field of ecology where discrete dynamical systems are often directly derived or assumed. **Cite.** There is a school of thought that advocates discrete-time dynamical systems to be more natural for modelling real-world observations than differential equations and we refer the interested reader to [?].

2.1 Invariant Sets

A core concept in the study of dynamical systems is that of invariance.

Definition 2 (Invariant Set). *Given a discrete-time dynamical system $T : U \rightarrow U$, a subset $A \subset U$ is said to be an invariant set if $T(A) = A$.*

We also define the orbit of a function T .

Definition 3 (Orbit of T). *The orbit of T is to be defined the sequence $\bar{u} = \{u_n\}_{n \in \mathbb{Z}}$ obeying the update equation, $u_{n+1} = Tu_n$, $n \in \mathbb{Z}$.*

Two examples of invariant sets include a fixed point where $Tu = u$ for $u \in U$, and a periodic orbit, i.e., a set of iterates $\{u, Tu, T^2u, \dots, T^pu\}$, where $T^{p+1}u = u$ for some $p \in \mathbb{Z}$. The entire space

U could be also be invariant. Consider for example the space $U = [0, 1]$, where $Tu = 4u(1 - u)$ and then U is invariant (as every $u \in U$ can be written as $u = 4x(1 - x)$ for some $x \in U$).

We may learn a great deal about the iterates of a dynamical system by considering the types of invariant sets of a discrete-time dynamical system. For example, if $U = [0, 1]$ has map $Tu = u/2$, then the only invariant set is $\{0\}$, and all orbits approach this invariant set as time flows in the forward direction. Indeed, if a some non-zero invariant set (call it B) exists, then there is some $r \in (0, 1] \cap B$. But $r \notin T(B)$ since any orbit with initial value r will be a decreasing sequence. Moreover, every orbit of T will be decreasing and therefore approach the value 0 as $n \rightarrow \infty$.

One may ask if every orbit approaches an invariant set? In general the answer is no, since for the dynamical system $T : \mathbb{R} \rightarrow \mathbb{R}$ defined by $Tu = 2u$, any orbit that does not intersect the invariant set $\{0\}$, will not approach any invariant set.

However, when the space U is compact, all orbits approach an invariant set. This is since the set of limit points of the orbit can be shown to be invariant [?]. So, when U is compact, the ω -limit set $\omega(u; T)$ of a point u defined to be the collection of limit points of the sequence $\{u, Tu, T^2u, \dots\}$ is nonempty, and $\omega(u; T)$ is invariant.

Example 1. For the map $Tu = u^2$ defined on $[0, 1]$ all orbits lie in the invariant set $\{1\}$ or else would approach the invariant set $\{0\}$.

Invariant sets have various properties. Invariant sets can be attracting or repelling depending on how orbits in its vicinity behave. Recall the examples $Tu = u/2$ and $Tu = 2u$ defined on \mathbb{R} (where $\{0\}$ “attracts” orbits) or the example, $Tu = u^2$ defined on $[0, 1]$ (where $\{1\}$ “repels” orbits). We are interested in attractive invariant sets since they capture the long-term dynamics as time increases. In particular we are interested in those invariant sets named attractors.

Definition 4 (Attractor). Let $T : U \rightarrow U$, where U is a metric space with metric d . A compact subset $A \subset U$ is said to be an attractor if it satisfies the three conditions:

1. A is invariant.
2. A is asymptotically stable, i.e., for every $\epsilon > 0$ and for all u so that $d(u, A) < \epsilon$, we have $d(T^n u, A) \rightarrow 0$ as $n \rightarrow \infty$.
3. A has Lyapunov stability, i.e., for every $\epsilon > 0$ there exists a $\delta(\epsilon) > 0$ so that $d(u, A) < \delta$ implies $d(T^n u, A) < \epsilon$ for all $n \geq 0$.

Indeed in our previous examples, it can be verified that the system $Tu = u/2$ defined on \mathbb{R} and $Tu = u^2$ defined on $[0, 1]$ the singleton set $\{0\}$ is an attractor. For the system, $Tu = 1 - |2u - 1|$ on $[0, 1]$, one may easily verify that the only attractor is the entire space $[0, 1]$. This is since between any two points $u < v$ in $[0, 1]$, and for any a, b so that $u \leq a < b \leq v$, we can find an n so that $T^n(a, b) = [0, 1]$ (as would be explained later in this chapter).

EdN:1 1

A dynamical system can have several attractors, and may also be contained in another attractor. For the example, $Tu = u^2$ on $[0, 1]$, both $\{0\}$ and $[0, 1]$ are attractors. It is a known result that for dynamical systems on a compact space there always exists an attractor.

The dynamics restricted to an invariant set can be complicated. For instance an invariant set could be just a single point or it could have an infinite set. If the invariant set is infinite then there is a possibility of complicated dynamics, and a particular well-studied phenomenon of such complexity gives rise to what is called chaotic behaviour, and has been studied in detail over the last fifty years.

2.2 Chaos

In the 1960s, a number of mathematicians and mathematically interested scientists independently discovered chaos in the mathematical sense. The meteorologist Edward Lorenz may have been the first to explain this phenomenon in his 1963 paper [?]. The notions of invariance, attractivity and chaos may also be described for continuous systems, and Lorenz's system comprised of a system of differential equations. The narrative of Lorenz's discovery of chaos, as well as the history of other forerunners in this subject, are fascinating. For those who are interested, we highly recommend James Gleick's book *Chaos: The Making of a New Science* [?]. It clearly illustrates these experiences, explains why chaos was such a startling and crucial mathematical and scientific discovery, and describes the underlying mathematical notions for non-specialists.

There are several intricately different definitions of chaos available in the literature (**Cite**), with each of them indicating some aspect of complexity. In practice, it is not possible to verify which definition or notion does the underlying dynamical system satisfies when only data is observed from it. We do, however, specifically recall the definition of chaos in the sense of Devaney [?, ?] so as to understand some nuances behind the complexity. Devaney's definition of chaos has three requirements, with the first being the notion of sensitive dependence on initial conditions.

Definition 5 (Sensitive Dependence on Initial Conditions). *A dynamical system $T : U \rightarrow U$ is said to have sensitive dependence on initial conditions (SDIC) if there exists a $\delta > 0$ such that for every $u \in U$ and in every neighborhood of $u \in U$ there exists a $v \in U$ and an integer $N := N(u, v) \in \mathbb{Z}$ such that $d(T^N u, T^N v) > \delta$.*

It is important to acquire a sense of this definition as it can easily be misunderstood. It is common

¹EDNOTE: For let B be an attractor of U . From above, we know that B is itself invariant, is asymptotically stable and has Lyapunov stability. Now suppose there is some attractor $C \subsetneq U = B$, so there is a $b \in U$ which is not in C
M: Have changed the example

misconception to interpret SDIC as two close points (u and v) that eventually become separated by a distance δ under iteration by T . But this is not true. To fully comprehend all the subtleties of the concept, one needs to discuss it more thoroughly by considering each sentence with care and attention, whereafter one may examine how they fit together to convey the concept of sensitive dependence.

To this end, we make 3 remarks:

1. First, the $\delta > 0$ in the definition of SDIC is independent of u .
2. Second, in every neighborhood of u , we may not necessarily find all points v in the neighborhood distinct from u that would separate from the forward iterates of u .
3. Finally, N depends upon u and v chosen, and their iterates may not separate for ever (i.e., for all $n > N$) and we allow their iterates to get arbitrarily close in the future.

To illustrate the concept of SDIC, we present two examples - one from Mathematics and the other from the field of Physics.

Example 2. *The logistic map(LM), a recursion relation of the form $x_{n+1} = rx_n(1 - x_n)$ where $x_n \in [0, 1]$ and $r < 4$ is of particular interest to us as it is classically used to illustrate the influence of chaos on a system (**Cite**). Below are plotted the first 75 iterates of the LM for $r = 3.7$ and $x_0 = 0.2$ with an adjustment of 0.001 in x_0 and then r .*

For an intuitive sense of the fact that the LM exhibits SDIC, we draw two green neighbourhoods of diameters $\delta_1 = 0.1, \delta_2 = 0.02$ around the points $u_1 = 0.7, u_2 = 0.4$ respectively. It is easy to see that every neighbourhood around these two points will contain some $v_1, v_2 \in [0, 1]$ and natural number $N_1, N_2 \in \mathbb{N}$ respectively s.t. the distance $d(T^{N_1}u_1, T^{N_1}v) > \delta_1 = 0.02$ and $d(T^{N_2}u_2, T^{N_2}v) > \delta_2 = 0.1$.

*In a more rigorous fashion*²

EdN:2

Example 3. *A second example pertains to a simple physical system called the double pendulum - a pendulum with another pendulum attached to its head given an initial position and angular velocity.*

Below are two figures denoting the trajectories of a double pendulum's second head after some elapsed time. Depicted are three pendulum heads with equal angular velocity, but differing ever so slightly in initial position. As can be seen below, the trajectories diverge very quickly to become completely different. The double pendulum will be discussed in much greater detail in a later chapter (5), but we mention the example here to provide a numerical example of a practical system exhibiting SDIC.

The second part in Devaney's definition of chaos concerns topological transitivity.

²EDNOTE: B:Should I be more rigorous here?

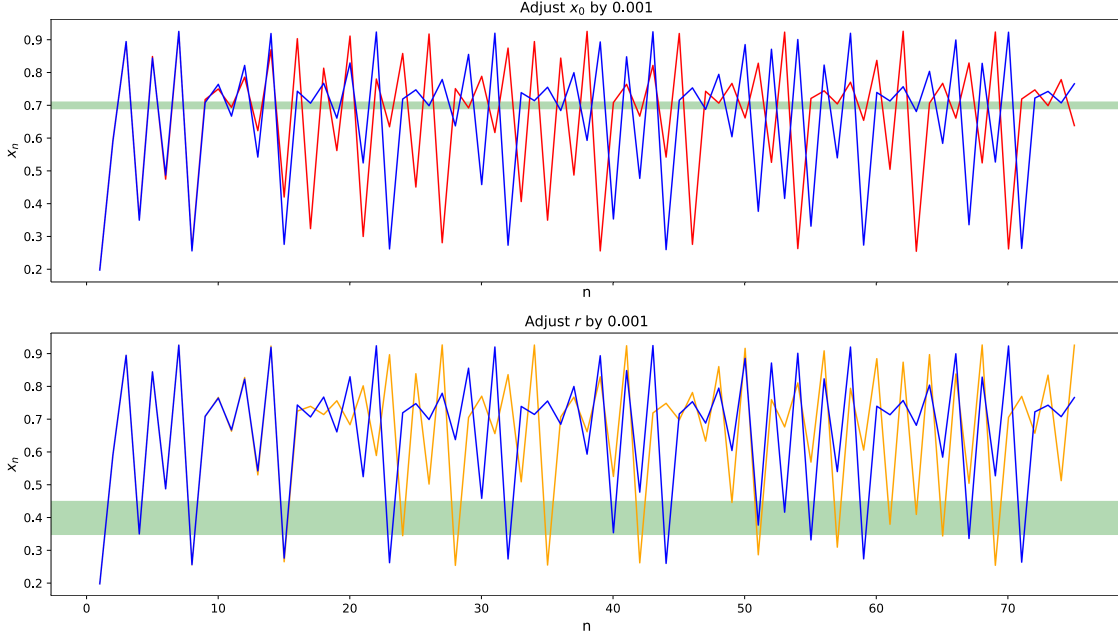


Fig. 1. Logistic Map for $r = 3.7$, $x_0 = 0.2$

Definition 6 (Topologically Transitive). A dynamical system $F : U \rightarrow U$ is topologically transitive if for any pair of nonempty open sets E_1 and E_2 there exists a $n \in \mathbb{N}$ such that $T^n(E_1) \cap E_2 \neq \emptyset$.

Topologically transitivity implies that the iterates of an open set of initial conditions gets mixed up with other open sets. On a compact metric space, it can be shown that topological transitivity also implies the existence of a point whose forward iterates are dense [?]; or in other words, the orbit going through this point will be dense in the compact metric space. In fact, it is topologically more likely that the choice of an arbitrary point will be one whose iterates are almost dense.

Definition 7 (Meager Set). A subset of a topological space U is said to be a meager set if it can be written as a countable union of sets with empty interior. The complement of the a meager set is set to be a residual set.

To be more precise (see [?]), for a discrete-time dynamical system on a compact space, the set of points with dense iterates are residual, and they are typical or likely to be observed in practice. In this thesis, we assume that when data is observed from a topologically transitive system, we assume that the data arises from a dense orbit.

We may now define the notion of Chaos as formulated by Devaney[?].

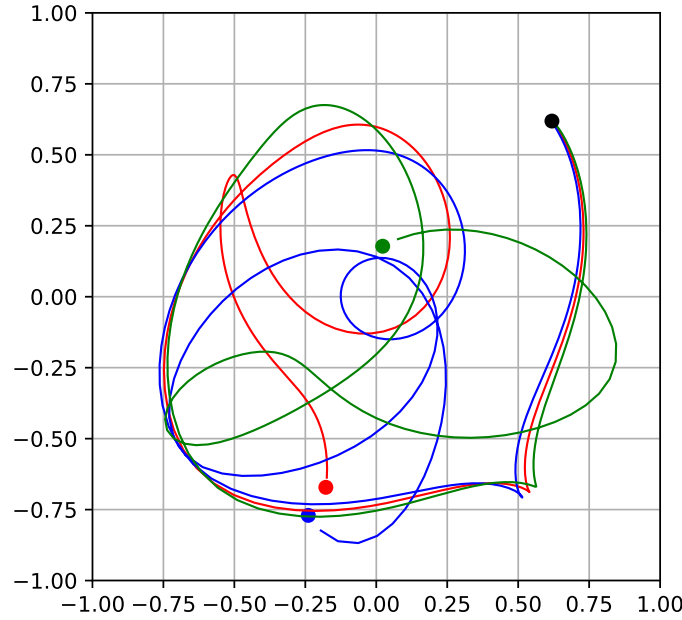


Fig. 2. Three double pendulum heads with initial angle differing by 0,025 radians

Definition 8 (Devaney's Chaos). A dynamical system $T : U \rightarrow U$ is said to exhibit chaos in the sense of Devaney if it satisfies the three properties:

1. T has SDIC.
2. T is topologically transitive.
3. The set of periodic points of T are dense in U .

Example 4. The standard tent map $Tu = 1 - |2u - 1|$ defined on $[0, 1]$ is a well-known example of a dynamical system satisfying these three properties.

We now reason as to why this is true. The graph of the map T is piecewise linear with two straight lines, one connecting the points $(0, 0)$ and $(\frac{1}{2}, 1)$ and the other connecting $(\frac{1}{2}, 1)$ with $(1, 0)$. They form a so-called tent with base centered at $u = 1/2$. The graph of the map T^2 comprises two symmetric tents with their base centered at $1/4$ and $3/4$. See Figure 3

In general, the graph of T^k contains 2^{k-1} tents. Given any point u and an open neighborhood $W \subset [0, 1]$ of u , we can find $k \in \mathbb{Z}$ large enough so that $T^k(W) = [0, 1]$ because we can accommodate a tent whose base is contained in W . This implies the existence of two points w_1 and w_2 in W so that $d(T^k w_1, T^k w_2) = 1$, and by the triangle inequality at least one of the inequalities $d(T^k u, T^k w_1) > \delta$ or $d(T^k u, T^k w_2) > \delta$ hold when $\delta \in (0, \frac{1}{2})$. So T has sensitive dependence on initial conditions.

Next, let W_1 and W_2 be two nonempty open sets. Given any open set W_1 , we can find a $k \in \mathbb{N}$

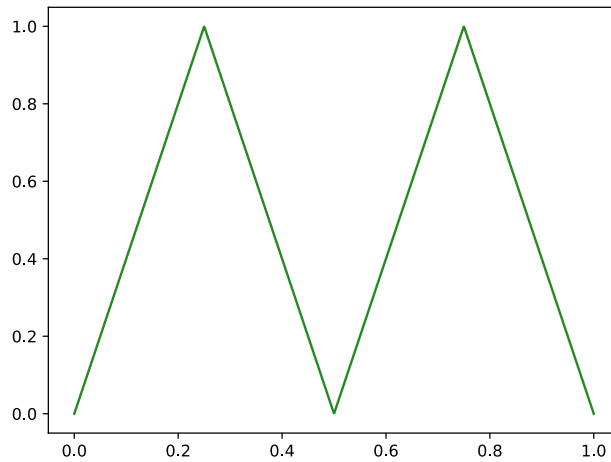


Fig. 3. Three double pendulum heads with initial angle differing by 0,025 radians

large enough so that $T^k(W_1) = [0, 1]$ because we can accommodate a tent with base contained in W . So $T^k(W_1) \cap W_2 \neq \emptyset$, and thus T has topological transitivity.

Finally, for every open interval (a, b) , the graph of T^k intersects the graph of the identity map on $[0, 1]$. This follows from the fact established above that there exists a $k \in \mathbb{N}$ such that $T^k(a, b) = [0, 1]$. If the intersection point has coordinates of the form (p, p) , the point p is then a fixed point of T^k and therefore also a periodic point of T . Since we have found a periodic point in the interval W , the set of periodic points are dense in T . This follows because any fixed point of T^k is also a periodic point, and we have found this in an arbitrary interval.

2.3 Conjugacy

We now turn our attention to the subject of conjugacy that describes when two dynamical systems are equivalent dynamically.

To show topological similarity or sameness between two metric or topological spaces, one needs to establish a homeomorphism between the two spaces. However, in the study of dynamical systems defined on two spaces, establishing a homeomorphism does not indicate that the systems are dynamically related in any way. For instance, the maps $Tu = u^2$ and $Tu = 1 - |2u - 1|$ defined on $[0, 1]$ have totally different behaviour. To find systems that are dynamically similar, one rather needs to establish a kind of dynamical equivalence which we illustrate in a simple commutativity diagram below.

$$\begin{array}{ccc}
 U & & U \\
 & \searrow & \searrow \\
 V & & V.
 \end{array} \tag{2.3}$$

To understand the diagram we note that if we travel “right and then down,” the diagram instructs us to use T (top arrow right) first, followed by ϕ . (right arrow downwards). Consequently the pathway amounts to finding $\phi(T(u))$. If we proceed ”down and then right”, the diagram instructs us to apply ϕ (left arrow downwards) first, and then apply S (bottom arrow right). This then amounts to finding $S(\phi(u))$. When the relationships in this diagram hold, we say $\phi(T(u)) = S(\phi(u))$, and we formally denote it as $\phi \circ T = S \circ \phi$.

Definition 9 (Conjugacy & Semi-Conjugacy). *Consider the dynamical systems $T : U \rightarrow U$, $S : V \rightarrow V$ and suppose the relationship $\phi \circ T = S \circ \phi$ holds. If $\phi : U \rightarrow V$ is a homeomorphism, then S is said to be conjugate to T and ϕ is called a conjugacy. If we relax the criterion on ϕ and merely require ϕ to be continuous and surjective where $\phi : U \rightarrow V$, then ϕ is a semi-conjugacy between T and S where T has domain U and S has domain V ; S is said to be semi-conjugate to T .*

³ When S is conjugate to T , the dynamics of the two systems are in some way ”dynamically equivalent”. Specifically, they are in one-to-one correspondence with one another. However when S is semi-conjugate to T with $\phi : U \rightarrow V$ a many-to-one mapping, the dynamics on V provide merely a coarse-grained description of the dynamics on U [?]. When S is semi-conjugate to T , it is also common to call S a *factor of T* , or conversely that T is an *extension of S* . In essence, an extension (e.g., [?]) is a larger system capturing all of the important dynamics of its factor. EdN:3

It is a very hard or nearly an impossible task to establish the existence of a conjugacy or a semi-conjugacy ϕ between two systems [?]. However, one can verify that a function ϕ satisfies the commutativity diagram 2.3.

NB NB FINISH

Example 5. *For example $\phi(x) = \sin(\frac{\pi}{2}x)^2$ is a conjugacy between two systems $Tu = 1 - |2u - 1|$ and $Sv = 4v(1 - v)$ defined on $U = [0, 1]$.*

To see this, we consider $\phi \circ T = \sin(\frac{\pi}{2}(1 - |2u - 1|))^2$

and $S \circ \phi = 4 \sin(\frac{\pi}{2}x)^2(1 - \sin(\frac{\pi}{2}x)^2)$

In establishing that two systems are conjugate (semi-conjugate) to one another, we have also shown that one may choose to work with one system as opposed to the second and still be guaranteed

³EDNOTE: M: Saying ϕ is then a semi-conjugacy between T and S could be ambiguous as it does not specify which system is on the top in the commutativity. So please edit accordingly. B: Does defining explicitly the domain and codomain of ϕ make it clear enough?

to obtain information on the latter. This is especially useful while forecasting the future evolution of dynamical systems.

Chapter 3

The Learning Problem

In this chapter we acquaint ourselves with the question of forecasting dynamical systems with unknown underlying structures, state and discuss the shortcomings of the Takens delay embedding theorem and discuss various issues faced while forecasting.

Consider a relatively simple learning problem: Given the sequence (u_0, u_1, \dots, u_m) for $m \in \mathbb{Z}$, a finite segment of an orbit of the map T where T is defined by the update equation $u_{n+1} = Tu_n$, forecast the values u_{m+1}, u_{m+2} where the map T is unknown, given that u_m .

A practical example of this would be the subsequent scenario: given the time-sequential coordinates of an object moving in space, predict the future positions of that object. However we are very rarely, if at all, presented with a problem where the entire state information is available to us in the form u_n at some time-step $n \in \mathbb{N}$. Consider then a more involved learning problem:

Suppose we only have the observations $\theta(u_0), \theta(u_1), \dots, \theta(u_m)$ of the true system states u in an unknown dynamical system (U, T) and we wish to predict the values $\theta(u_{m+1}), \theta(u_{m+2})$ and so forth.

First we establish the method of Takens delay embedding. In this method, one considers a discrete-time dynamical system defined on a 'nice' space (a smooth manifold)⁴ that can be obtained as time- K map of a continuous time-dynamical system⁵. We make observations from such a system, i.e., we observe the evolution of $\theta(w_0)$ (where w_0 represents the initial state) i.e., we observe a finite set of values of $u_n := \theta(w_n) = \theta(Tw_{n-1})$. The sequence $\{u_n\}$ represents a scalar time-series and intuitively θ may be thought to represent a probe inserted into a larger system which is itself only measuring/extracting a small part of the greater system state at time $t = n$. Consider for example a thermometer erected to measure the ambient temperature in a local village. This measurement

⁴EdNOTE: Should this be further explained?

⁵EdNOTE: B: Should one add here: "These notions will be made formal below"?

function, the thermometer, is capturing only a single aspect, the temperature, of a much grander dynamical system entailing the present weather of the surrounding area. Even more than that though- it is measuring a miniscule part of the global weather system.

However, the true state u of a system is seldom, if ever, fully known. In almost all cases we can at most insert probes into a system to obtain partial information by means of the measurements taken. Moreover, the process of taking a measurement itself introduces 2 additional aspects which complicate the problem: ⁶

EdN:6

- i. A series of measurements over a specific time-interval is inherently a discretisation procedure of the underlying continuous-time dynamical system. (Hence why we restricted our attention to discrete-time systems in the preceding section)
- ii. A probe will never be fully accurate, and so the act of measurement introduces a certain measure of numerical noise/inaccuracy.

From here we construct a multidimensional observable using the method of stacking previous observations, i.e., we create delay-coordinate map defined by $\Phi_{k,\theta}(w) := (\theta(T^{-k}w) \dots, \theta(T^{-1}w), \theta(w))$. The essence of Takens theorem relates to the fact that when k is sufficiently large, we can define a dynamical system on the space \mathbb{R}^{k+1} whose states are $\Phi_{k,\theta}(w)$, $\Phi_{k,\theta}(Tw)$, $\Phi_{k,\theta}(T^2w)$, etc. and this dynamical system is topologically conjugate to the unknown underlying system (W, T) . We recall the Takens delay embedding theorem next.

3.1 Taken Embedding Theorem

We define first the concepts of homeomorphism and embedding:

Definition 10 (Homeomorphism). *A homeomorphism is a function $f : Z \rightarrow Y$ between two topological spaces Z and Y that is continuous, bijective and has a continuous inverse.*

Definition 11 (Embedding). *Consider a homeomorphism $f : Z \rightarrow Y$ for $Y \subset X$. Z is said to be embedded in X by f .*

Takens Theorem states a result establishing a relationship between the observed and underlying dynamical systems by showing that the concatenation of a sufficiently large number of previous observations into a vector will, under certain conditions, generate a map between the vectors from the respective systems. We formulate the theorem from [?].

Theorem 1 (Takens Embedding Theorem (adopted from [?]). *Let W be a compact manifold of dimension m , and $d \geq m$ so that $2d$ is an integer. It is a generic property for the pair (T, θ) ,*

⁶EdNOTE: M: Maybe replace difficulty by saying that we need to consider two aspects. B: Perhaps this edit makes more sense?

where $T : W \rightarrow W$ is a smooth diffeomorphism, and $\theta : W \rightarrow \mathbb{R}$ a smooth function, the map $\Phi_{2d,\theta} : W \rightarrow \mathbb{R}^{2d+1}$ defined on W by $\Phi_{2d,\theta}(w) := (\theta(T^{-2d}w), \dots, \theta(T^{-1}w), \theta(w))$ is a diffeomorphic embedding; by ‘smooth’ we mean at least C^2 . Consequently, there exists a map $F_\theta : \Phi_{2d,\theta}(W) \rightarrow \Phi_{2d,\theta}(W)$ defined by

$$F_\theta : (\theta(T^{-2d}w), \dots, \theta(T^{-1}w), \theta(w)) \mapsto (\theta(T^{-2d+1}w), \dots, \theta(w), \theta(Tw))$$

so that (W, T) is topologically conjugate to $(\Phi_{2d,\theta}(W), F_\theta)$.

By generic we mean a residual set on a certain topology on appropriate spaces of functions (that we do not describe here) (**Cite**). By C^2 , we make reference to a twice-differentiable function with a continuous second derivative. In our scenario (and this remains important throughout) the input space U is considered the attractor.

Below we reproduce Figure 1 from [?].

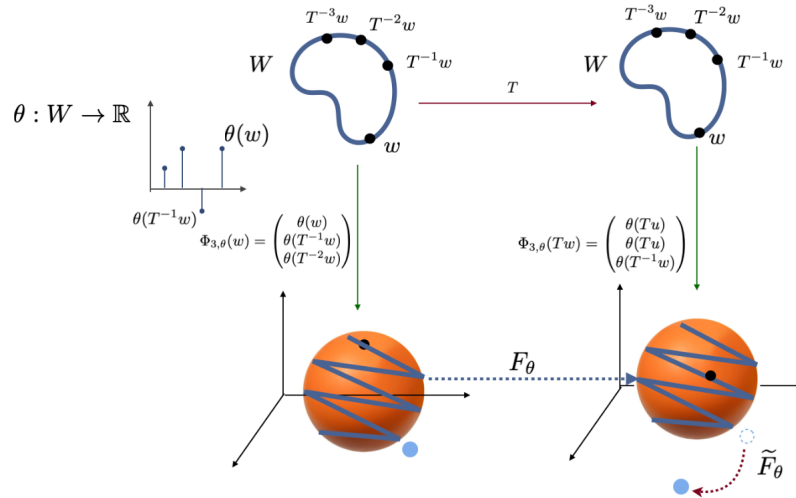


Fig. 1. Schematic of Takens Embedding Theorem's usage and consequence of how iterates ...

7 8

EdN:7

EdN:8

If we can learn the map F_θ , from sufficient number of data points of $\{\phi_{2d,\theta}(w_n)\}$, then we also know how to forecast how $\theta(w_n)$.

⁷EDNOTE: M: Here, and throughout if you reproduce a figure, it must be stated where it was reproduced from. B: How does one write that a figure is reproduced? Also, should \tilde{F} be left out for now?

⁸EDNOTE: B: Could I perhaps have the original figure? This is only a screenshot used as a temporary placeholder.

3.2 Practical Issues of Takens theorem

It is a fact that Takens' Embedding Theorem is both a powerful result and that it provides compelling reason to believe that one could conceivably reconstruct accurately a system conjugate to the underlying system. Nonetheless, it does present some serious practical limitations:

1. Even supposing that we could indeed find F_θ , our *approximation* of F_θ is a map from a larger set \mathbb{R}^{2d+1} containing the embedded attractor. There are, however, no theoretical guarantees that F_θ will retain $V = \Phi_{2d,\theta}(W)$ as an attractor although W could itself be an attractor.
2. Takens theorem is stated for noiseless observations. Due to noise ϵ_n the delay vector $\Phi_{2d,\theta}(w_n) + \epsilon_n$ may lie outside V . Furthermore, due to the chaotic nature of the underlying system (i.e. the fact that it has SIDC), the evolution of $\Phi_{2d,\theta}(w_n) + \epsilon_n$ under the map F_θ could move out of V completely. This problem can be overcome by using a driven dynamical system with some properties, and we discuss this in Chapter *.

Continue writing here

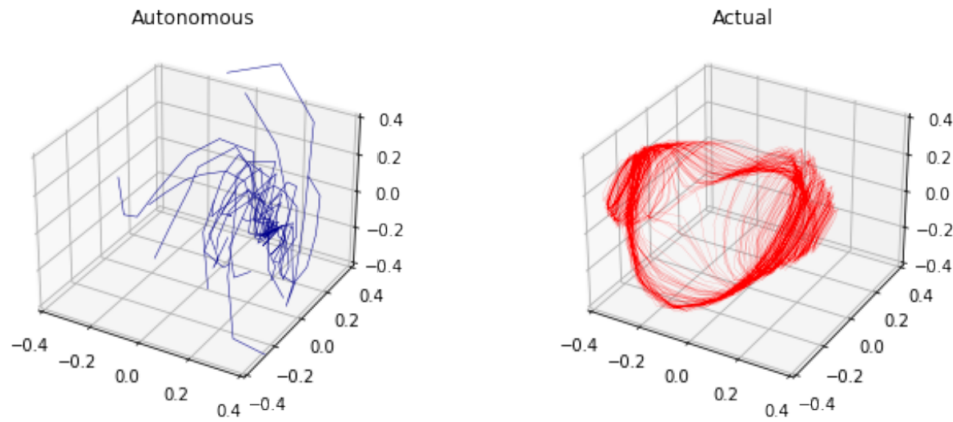


Fig. 2. Illustration of a system having failed to learn the map

Explanation continued here

Continue explaining here

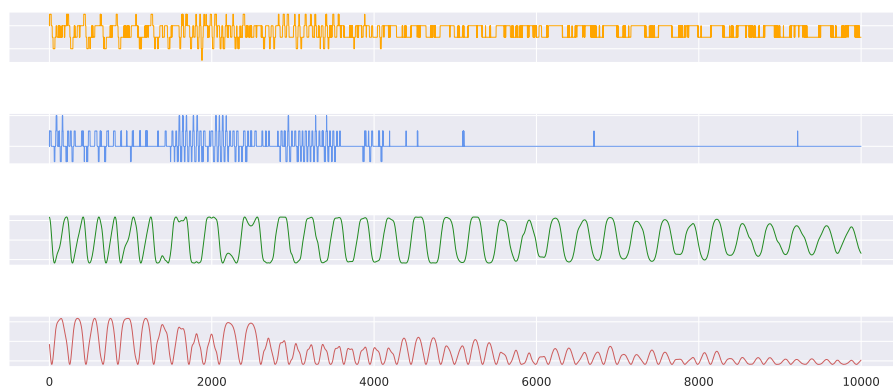


Fig. 3. A damped pendulum

Chapter 4

Driven Dynamical Systems to Forecast Problems

In this chapter, we discuss results pertaining to the mapping of temporal data obtained from a discrete-dynamical system onto a different space through the notion of a driven dynamical system. We also consider the conditions a driven system should possess to avoid adding distortion to its state-space representation. We then describe the *single-delay dynamics* (SDD) of the system and and finally recall conditions on the driven system so that the SDD are conjugate (or at least semi-conjugate) to the underlying system. The SDD can then be used to forecast and reconstruct the underlying system.

4.1 Nonautonomous and Driven Dynamical Systems

Definition 12 (Nonautonomous Dynamical System (adopted from [?])). A nonautonomous dynamical system (NDS) on a space X is simply a dynamical system comprising a family of maps $\{g_n\}_{n \in \mathbb{Z}}$ where $g_n(\cdot) := g(u_n, \cdot) : X \rightarrow X$ is continuous. Each g_n arises as a consequence of an input u_n from the input space U , a topological space.

We immediately recall the figure first defined in (2.3) and remind ourselves of our ultimate goal to learn a system that is conjugate to an underlying or unknown dynamical system (U, T) by making use of the measurements obtained from the unknown system.

$$\begin{array}{cc} U & U \\ V & V. \end{array}$$

To find a function ϕ , the need now arises to consider a so-called driven dynamical system, a special case of a nonautonomous dynamical system, where two spaces are considered: input is taken from an input space U and the state of a space X is updated. (Previously, only the state was considered.) We do this to mimic the true conditions. By accounting for both external and internal factors which influence the system-state at a specific time-step, our model resembles nature much more realistically.⁹

EdN:9

Definition 13 (Driven, Compactly Driven Dynamical System). A driven dynamical system comprises two topological metric spaces U , X and a continuous function $g : U \times X \rightarrow X$ where $g(u_n, x_n) = x_{n+1}$. If the input space U is compact, we refer to the system as compactly driven.

The dynamics on X are generated by the update equation $x_{n+1} = g(u_n, x_n)$ where $n \in \mathbb{Z}$, input u_n from U and state x_n belonging to X , where X is compact. Abbreviated, we shall refer only to the *driven system* g , with all other entities being understood implicitly.

Notably, a nonautonomous dynamical system may be generated from U ; any input \bar{u} , a bi-infinite sequence from U , gives rise to the sequence of self-maps $\{g(u_n, \cdot)\}_{n \in \mathbb{Z}}$ contained in X . Physically, one may think of this bi-infinite sequence as referring to a system that has been running for a long period at the time of the first measurement taken from the system (alternatively, the first time a probe is inserted into the system to take an observation)

Definition 14 (Entire Solution). A sequence $\{x_n\}_{n \in \mathbb{Z}} \subset X$ is called an entire solution (or simply a solution) of the driven system g with input \bar{u} when it satisfies

$$g(u_{n-1}, x_{n-1}) = x_n$$

for all $n \in \mathbb{Z}$

It is important to emphasise that a sequence $\{x_n\}$ satisfying the update equation above can only be a solution if $x_n \in U$ holds for all $n \in \mathbb{Z}$ and just for $n > 0$. Consider the example below.

Example 6. The only solution $\{x_n\}_{n \in \mathbb{Z}}$ to the driven system $g(u, x) = \frac{ux}{2}$, where $X = [0, 1]$, $U = [0, 1]$, is the zero solution $x_n \equiv 0$. To see this, consider any $x_n = a \in [0, 1]$ where $a \neq 0$. Let $\bar{u} \in U$ be a non-zero constant sequence, say $u_n = 0.5$. The driven system may be rewritten as $x_{n-1} = \frac{2x_n}{u_{n-1}} = 4x_n$ and the iterates of x_n in backward time will increase by a factor of 4 at each timestep. Thus for some $m \leq n$, $1 < x_m$ i.e. $x_m \notin X$. So $\{x_n\}_{n \in \mathbb{Z}}$ is not a solution and it follows that the only possible solution is the zero solution.

⁹EDNOTE: M : I understand this sentence, but am not sure if a general audience would. So please elaborate or omit.
B: Is this satisfactory?

A system may also have multiple solutions as is evidenced in the example below.

Example 7. Consider the driven system $g(u_n, x_n) = x^n$ for $X = [0, 1]$, $U = \mathbb{R}$. The system has an uncountable number of solutions, as there exists a solution for every $x \in X$ which also passes through the point x and $\lim_{n \rightarrow \infty} x_n = 0$, $\lim_{n \rightarrow -\infty} x_n = 1$. The proof is deferred to immediately after the next paragraph (see 1).

As the solutions to a driven system are often considered to be an important entity [?], we next identify a subspace X_U of X that contains all possible solutions. To realize such a subspace of a driven system g , the concept of a reachable set is defined.

Definition 15 (Reachable Set). The reachable set of a driven system g is exactly the union of all the elements of all the (entire) solutions, i.e.,

$$X_U := \left\{ x \in X : x = x_k \text{ where } \{x_n\} \text{ is a solution for some } \bar{u} \right\}.$$

The set of all reachable states at a specific time n for input \bar{u} is denoted by $X_n(\bar{u})$

The reachable set itself can be defined independently from the fact that g does or does not have the ESP/USP. We note that $x \in X_n(\bar{u})$ if and only if g has a solution $\{x_k\}$ for $x_n = x$ and input \bar{u} . (Cite).

X_n is exactly the set of points x through which some entire solution Ψ passes at time n .

This will lead to a result established later, but which is worth taking note of now: *g being a topological contraction is equivalent to the existence of a unique entire solution.*

Remark 1 (Proof of Example 7). To see this, we show that for every input $\bar{u} = \{u_n\}_{n \in \mathbb{Z}}$, $g(u_n, \cdot)$ is a contraction map on X for every $k \in \mathbb{Z}$. Indeed $g(u_n, x_n) = x^n$ depends only on the state at time n and so for each input \bar{u} , $X_n(\bar{u}) = \{x^n\}$ is a singleton subset. We then easily conclude that $\{x_n\} \equiv 0$ is the only solution.

Thus far it has been demonstrated that a system may have one or more solutions; one may ask if a driven system always has a solution and, if so, whether it satisfies certain properties such as uniqueness. Should the driven system be compact, existence follows immediately as shown in the following result.

Theorem 2. *Let g be a driven system. If X is compact then for each input \bar{u} , there exists at least one solution to the driven system $g(\cdot, x)$*

¹⁰EDNOTE: B: Is this sufficient?

Proof. May be found in [?, ?, ?]

□

One may easily construct many systems with trivial solution-sets, such as $g(u, x) = x$ which has only the constant solution x and so for $U = [-1, 1]$, the system would have no solution if $|x| > 1$. To refine the scenario, we consider only systems with unique solutions.

4.2 Unique Solution Property

Definition 16 (Unique Solution Property). A driven system g is said to have the Unique Solution Property (USP) if for each input \bar{u} there exists exactly one solution. Alternatively we may formulate the USP as follows: g has the Unique Solution Property if there exists a well-defined map $\Psi : U \rightarrow X$ with $\Psi(\bar{u})$ denoting the unique solution.

One of the first result obtained upon defining the USP, is the fact that every solution will attract any different initial conditions towards the component parts of the solution. If g has the USP, then any solution to g is a uniform attractor in the nonautonomous dynamical systems literature [?]. The discussion on nonautonomous attractors is beyond the scope of this thesis and we refer the reader to [?, ?] for further reading.

NEED TO PARAPHRASE

In our context, g having the USP is equivalent to g being a topological contraction [?], or alternatively that g exhibits the Echo State Property(ESP). In [?] the echo state property is a key for the design and training of recurrent neural networks within the paradigm of reservoir computing. Available definitions and theorems concerning the echo state property, however, are of little practical use because they do not relate the network response to temporal or statistical properties of the driving input.

A greater discussion is presented in [?] a particular stability concept for which a number of equivalent definitions are available [?].

In most of the reservoir computing (RC) literature, the idea of memory-loss or fading memory popularly called the echo state property (ESP) is treated as a stability property of the system valid for all inputs. In such a case, the ESP guarantees that the entire past history of the input determines the state of the system precisely, i.e., there is no possibility of two or more reachable states at any given point in time [5] if the entire past values of the input have influenced the dynamics. The formal definition linking ESP intrinsically to an input is available in [4].

In the context of information processing, when a driven system g has a causal embedding, it helps carry over any input \bar{u} onto the solution space without distortion or loss of information.

The question now becomes whether we may establish a semi-conjugacy as presented below for the driven system g .

$$\begin{array}{ccc} \overleftarrow{U} & & \overleftarrow{U} \\ & \searrow & \searrow \\ X_U & & X_U. \end{array} \quad (4.1)$$

Restricting our attention more and more, we now consider a specific subclass of conjugacies.

Definition 17 (Universal Semi-Conjugacy). *Given a driven system g , we call a continuous and surjective map $h : \overleftarrow{U} \rightarrow X_U$ a universal semi-conjugacy if diagram 4.1 commutes for all $v \in U$.*

If the universal semi-conjugacy h exists (i.e. the diagram in 4.1 commutes) then the solution Ψ will be at least a coarse-grained representation of the input u .

Does such a function h for the driven system defined above exist? If g has the USP and $\Psi(u) = \{x_n\}_{n \in \mathbb{Z}}$ then h , defined by $h(\tilde{u}_n) := g(u_n, x_{n-1}) = x_n$, will satisfy the semi-conjugacy in the graph above 4.1. Regrettably, such a continuous mapping h is not guaranteed to exist [?] when g does not have the USP [?, Lemma 5]. Note that even when h does exist, we are not guaranteed its injectivity. Considering again example 6, we see that even if h were to exist, it could not be injective as $X_U = \{0\}$.

Re-sketching the commutativity diagram 4.1 above by fixing the input v in g and replacing X_U by its left-infinite sequence space \overleftarrow{X}_U , we obtain the diagram below. In this case, the function $H : \overleftarrow{U} \rightarrow \overleftarrow{X}_U$, a map that is both continuous and surjective, is called a *causal mapping*.

make special note of a

Definition 18 (Causal Mapping). *A continuous, surjective map $H : \overleftarrow{U} \rightarrow \overleftarrow{X}_U$ such that*

$$H \circ \tilde{g}_v = \sigma \circ H$$

where \tilde{g}_v maps (\dots, u_{-2}, u_{-1}) to $(\dots, u_{-2}, u_{-1}, g(v, u_{-1}))$.

$$\begin{array}{ccc} \overleftarrow{U} & & \overleftarrow{U} \\ & \searrow & \searrow \\ \overleftarrow{X}_U & & \overleftarrow{X}_U. \end{array} \quad (4.2)$$

Theorem 3. *For a compactly driven system, a causal mapping H exists if and only if g has the USP.*

Proof. See [?, Th.3]. □

When considering the map H , we once again restrict our focus of H to the function $H_2(\bar{u}) := (h(r\bar{u}), h(\bar{u}))$.

An embedding of the space \overleftarrow{U} in \overleftarrow{U} would allow one to establish topological conjugacy, which in turn provides stronger results than merely obtaining a coarse-grained representation via a causal mapping [?, ?].

Before formalising this, we make mention of the concept of the inverse-limit system of a dynamical system (U, T) . These spaces are noteworthy as their topology can help one ascertain much about the original system [?]. The topological structure for inverse-limit systems of chaotic dynamical systems is complicated, so we relegate further discussion to [?, ?]. In broad sweeps, the inverse-limit space of a dynamical system is a self-map on a subset of an infinite-dimensional space where each point in the inverse-limit space corresponds to a backward orbit of the map T . Following [?, ?]:

Definition 19 (Inverse-Limit Space, Inverse-Limit System). The subset $\widehat{U}_T \subset \overleftarrow{U}$ defined by

$$\widehat{U}_T := \{(\dots, u_{-2}, u_{-1}) : Tu_i = u_{i+1}\}$$

is called the inverse-limit space of (U, T) . The self-map \widehat{T} induced by T on \widehat{U}_T is defined by

$$\widehat{T} : (\dots, u_{-2}, u_{-1}) \mapsto (\dots, u_{-2}, u_{-1}, T(u_{-1}))$$

and the resulting dynamical system $(\widehat{U}_T, \widehat{T})$ is called the inverse-limit system of (U, T) .

Definition 20 (Causal Embedding). A driven system g is said to causally embed the dynamical system (U, T) if

- i. The diagram in 4.1 commutes (i.e. h is a universal semi-conjugacy),
- ii. $H_2(\bar{u}) := (h(r\bar{u}), h(\bar{u}))$ embeds the inverse-limit space $(\widehat{U}_T, \widehat{T})$ of (U, T) in $X \times X$.

The driven system g can induce an embedding of \overleftarrow{U} in \overleftarrow{X} as follows: If $H : \overleftarrow{U} \rightarrow \overleftarrow{X}_U$ is also injective (in addition to being surjective, and hence bijective), it becomes the embedding of \overleftarrow{U} in \overleftarrow{X} induced by the driven system g and we refer to H as a *causal embedding*¹¹.

Theorem 4. *If $g(\cdot, x)$ is invertible and has the USP then H is a causal embedding.*

Proof. See [?]. □

When g has the USP, the diagram below illustrates the operation of the mappings h and H . The mapping $h : \overleftarrow{U} \rightarrow X_U$ is an observable as discussed in an above section (??).

¹¹EDNOTE: M : Causal embedding a dynamical system is different from the definition of causal embedding, so please edit/delete this paragraph and the next theorem and also in the remaining part till Section 1.2

The next result is of relative import but to fully comprehend its influence, some discussion still remains. We state it in part here.

Theorem 5. *The following statements are equivalent:*

- i. g has the USP.
- ii. g is a topological contraction.

4.3 Choosing the driven system g

One must be careful to avoid a choice of g which would add complexity to the obtained solution.

When a causal embedding H exists for the driven system g , one can map an arbitrary input u onto the solution space X without additional distortion or information-loss (**Cite.**). When an embedding is established, the question of possible additional complexity in the solution is removed by guaranteeing that, since the systems are conjugate (semi-conjugate, (**refer**)), g does not add any (some) complexity to the system.

It is, however, a balancing act as it also undesirable to choose a function g that quenches the temporal structure in u by contracting to such a degree that the ability to recover information from the original system is lost completely. In the above example (6), the input's temporal variation cannot be related to the reachable set since X_U consists of a single element; little, if not no, information is encoded. To obtain a suitably complex function g , it is thus desired that the reachable set of a driven system be large enough to relate to the input. We require that the reachable set of a driven system to be such that the inverse-limit space of U_T can be embedded in some finite self-product of the reachable set of T (**refer or cite**). To this end, consider the notion of State-Input (SI) Invertibility.¹²

EdN:12

Definition 21 (SI-Invertibility). A map g is said to be SI-Invertible if g is invertible for all $x \in X$. Alternatively it may be said that if, given x_n and x_{n-1} , u_n can be uniquely determined from $x_n = g(u_n, x_{n-1})$, then g is said to be SI-invertible.

SI-invertibility promises that 'enough' information is retained when g is chosen without introducing additional unwanted complexity.¹³ 'Enough' here refers to the fact that we may always recover the previous input value u_n (if we know successive states x_{n-1}, x_n). 'Without additional complexity' is guaranteed by the invertibility of the function g that guarantees a one-to-one mapping between u_n, x_{n-1} and x_n . In obtaining an SI-invertible function, therefore, we are guaranteed a system which will not lose so much information about its previous states in the forward-flow of time that one may

EdN:13

¹²EDNOTE: (M: Expand here. It stops a bit abruptly. B: Does the above reorganisation read better?)

¹³EDNOTE: (M: Expand why this is true. B: Is this satisfactory?)

not make any credible claims as to the original system's behaviour and properties. Simultaneously, it does guarantee us that the important information encoded in previous inputs is indeed preserved.

It is worth at this stage taking note of a specific driven system in the form of a discrete state-space model that has acquired some adherence in applications [?] and especially those pertaining to Echo State Networks and ESP. The function

$$g(u, x) = (1 - a)x + a\overline{\tanh}(Au + \alpha Bx) \quad (4.3)$$

is both SI-invertible and, if $\alpha\beta < 1$, also possesses the USP. One may easily show that g is SI-invertible by recovering u_n by

$$u_n := A^{-1} \left(\overline{\tanh}^{-1} \frac{1}{a} (x_{n+1} - (1 - a)x_n) \right) - \alpha Bx_n \quad (4.4)$$

for x_{n-1}, x_n known.

The proof that the map g possesses the USP is rather involved and hence, instead of being replicated here, may be found in [?, Th.2] This specific driving function g is used in our implementation and is discussed more completely in chapter ??

Despite the ease that one may work with a left-infinite history in the realm of theory, it is impossible to obtain such a sequence in any real-life application. One does not in practice, fortunately, need the entire left-infinite history of the input thanks to the Uniform Attraction Property (UAP).

Definition 22 (Uniform Attraction Property). A driven system g has the Uniform Attraction Property (UAP) if, regardless of starting position, all trajectories (**of what?**) converge to a single trajectory as time flows forward. This trajectory is also the unique solution sequence x to the input sequence u as mentioned above ??.

This definition is stated in a not completely rigorous manner as the more formal definition makes use of processes, a concept which would take some time to establish and detracts from the principal thrust of this project. ¹⁴

As the Unique Attraction Property guarantees that all trajectories will converge to the same trajectory as time moves forward. Incredibly, this permits one to initialise a driven system g with an altogether arbitrary initial value $y_m \in X$ where $m \in \mathbb{Z}$ and the UAP then guarantees that the sequence $\{y_{m+1}, y_{m+2}, \dots\}$ which satisfies the relation $y_n = g(u_n, y_{n-1})$ for $k \geq m$ will uniformly approach the elements $\{x_n\}$ of the actual solution (See [?, Th.11,] or [?, Eqn.18]).

One may now appreciate an even more astounding result: g having the USP is equivalent to the UAP. We restate the above theorem.

¹⁴EDNOTE: B: Is it very obvious to the reader at this point what the general thrust of the paper is?

Fig. 1. Approaching the true solution if g has the UAP

Theorem 6. *The following statements are equivalent:*

- i. g has the USP.
- ii. g is a topological contraction.
- iii. g has the UAP.

Proof. May be found in [?, Th.1] □

One need therefore not establish any additional results to ensure that the sequence uniformly approaches the unique solution Ψ . This vastly simplifies the effort necessary to set up a problem in order to guarantee that the underlying system will be accurately represented by the conjugate system.

This also solves the problem of perturbation/noise introduced by the observable or measurement function. Recall from before that an observable is inherently a map that discretises the underlying continuous-time system. Moreover, measurement and device error introduce mistakes. Since we are working with a chaotic system displaying sensitive dependence on initial conditions, these small errors could potentially send the trajectory into a completely different attractor. In the presence of the UAP, however, small measurement errors introduced into the system do not pose the same danger as before since the trajectories will eventually tend towards the actual solution. ¹⁵

EdN:15

4.4 The next step in Dynamics

Next we define the relation Y_T induced by (U, T) on $X_U \times X_U$ for a driven system g possessing SI-invertibility. To describe the single-delay lag dynamics formally, we consider a dynamical system $T : U \rightarrow U$ and we define a relation on the reachable set X_U , i.e., a subset defined on $X_U \times X_U$ by

$$Y_T := \{(x_{n-1}, x_n) : \{x_k\}_{k \in \mathbb{Z}} \text{ is a solution for some orbit of } T \text{ and } n \in \mathbb{Z}\}.$$

The following theorem establishes the existence of a well-defined map G_T describing the single-delay dynamics(SDD) of the system above.

Theorem 7. *If we let $G_T : Y_T \rightarrow Y_T$ be a map defined by the relation $(x_{n-1}, x_n) \mapsto (x_n, x_{n-1})$, then G_T is well-defined (and this results holds even in the absence of g possessing over the USP)*

¹⁵EDNOTE: (B: Complete enough?)

We're now getting quite close to where we want to be and our results carry more and more weight. Recall that g is only being given inputs from the orbits of T .

Definition 23 (Inverse-Limit Space). The inverse-limit space \widehat{U}_T , a subspace of \overleftarrow{U} , is defined by

$$\widehat{U}_T := \{(\dots, u_{-2}, u_{-1}) : T(u_n) = u_{n+1}\}$$

The inverse-limit space is well-defined since $T : U \rightarrow U$ is surjective by assumption.

Theorem 8. *Graph 4.1 is exactly the inverse-limit system $(\widehat{U}, \widehat{T})$.*

Note that H_2 maps an entire left-infinite solution sequence from Ψ to an element in $X \times X$. We now have the following (compare with 4.2 above):

NB NB NB - Causal Embedding Theorem! - Th.4

$$\begin{array}{ccc} \widehat{U}_T & & \widehat{U}_T \\ & Y_T & Y_T \end{array} \quad (4.5)$$

Theorem 9. (Y_T, G_T) is semi-conjugate to $(\widehat{U}, \widehat{T})$.

Proof. See [?, Th.3, Th.4] □

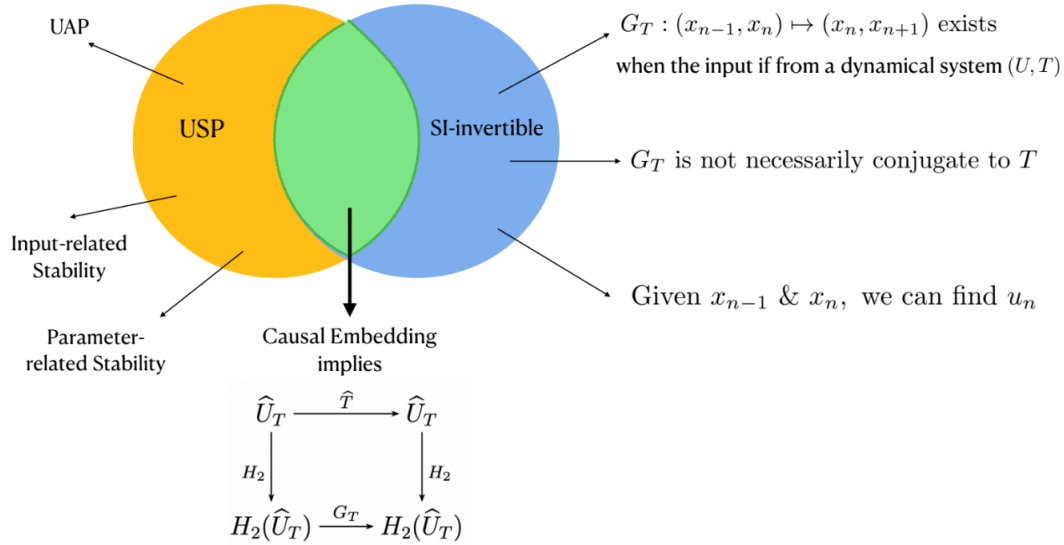
Summarising the discussing thus far:

It is easy to lose the birds-eye view, so we take a moment to review our progress up until this point.

1. We are interested in a some dynamical system (U, T) with unknown dynamics.
2. To determine properties about this system (U, T) and predict its future evolution, we determine the dynamics of the inverse-limit system $(\widehat{U}, \widehat{T})$. Given certain assumptions, we can guarantee that $(\widehat{U}, \widehat{T})$ is at least semi-conjugate to (U, T) .
3. If the driven system g is SI-invertible (and $\{u_n\} \in U$ is an orbit of T), the map G_T exists.
4. If, furthermore, g has the USP, (Y_T, G_T) is semi-conjugate to $(\widehat{U}, \widehat{T})$.
5. If we can assume that T is a homeomorphism, (Y_T, G_T) is topologically conjugate to $(\widehat{U}, \widehat{T})$, an extension space of (U, T)

One can therefore learn the SDD of the driven system states via G_T with enough data thanks to the USP/UAP. This enables us to do at least 2 things:

- Forecast future values of x_n via iterates of G_T (as G_T can be determined).
- Forecast future values of u_n .



4.5 A discussion of G_T

In the above sections, we established the map G_T describing the SDD of a driven system. By establishing the existence of this map, we've essentially embedded the attractor U (recall from 3.1) into the higher dimensional space $X \times X$. In layman's terms, this ensures that there is more "dimensional room" for the underlying system's underlying dynamics to "move". Since the dynamics aren't as "squashed", we might therefore hope that the dynamics of G_T are in some sense simpler than that of T . (Taken note of the fact that G_T is a homeomorphism even when T is just continuous)

In [?] it is illustrated in an empirical fashion in that the map G_T describes dynamics which are less functionally complex than that of T or of the map $\Phi_{2d,\theta}$. This is done by implementing a Recurrent Neural Network (RNN), which is discussed in 5. ¹⁶

EdN:16

We opt to learn G_T in an indirect manner by defining a new map $\Gamma : (x_{n-1}, x_n) \mapsto u_n$. It will follow immediately from G_T 's existence that Γ also exists.

¹⁶EDNOTE: (B: Should I add in that we'll be using Pearson coefficient?)

The reasons for taking this roundabout approach remain to be discussed in 4.6, but a pat answer may immediately be given: When Γ has been learnt, the system can be driven autonomously and then G_T is known anyway. We formalise this with a theorem.

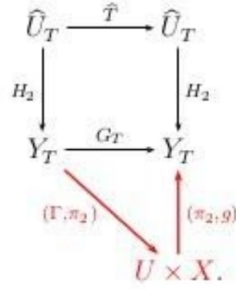
Theorem 10. *When x_n and x_{n-1} are successive points on a solution obtained for input-orbit of T $\{u_n\}$, then the map $\Gamma : X \times X \rightarrow U$ defined by $(x_{n-1}, x_n) \mapsto u_n$ exists whenever G_T exists*

Proof. See [?, Th. 3c]. □

Projection mappings π_i are defined in the traditional meaning where a k -dimensional vector is projected to it's i^{th} component such that

$$\pi_i : (a_1, a_2, \dots, a_{k-1}, a_k) \rightarrow a_i$$

The graph in equation 4.5 is then extended as below:



The problem finally simplifies to the issue of learning the map Γ and combining this with the projection mapping π_2 and the function g , which will be known. A final set of equations is obtained - equations that have been entirely constructed from data.

$$u_{k+1} = \pi_1 \circ (\Gamma, \pi_2) \circ (\pi_2, g)(u_k, x_k) \quad (4.6)$$

$$x_{k+1} = \pi_2 \circ (\Gamma, \pi_2) \circ (\pi_2, g)(u_k, x_k). \quad (4.7)$$

4.6 Advantages of learning Γ

One may immediately ask why we opt to take such a roundabout route; why not just learn the map G_T from the get-go? On the surface, this seems to be an arbitrary decision path with no real reasoning, so we take a pause again and discuss the motivation for learning Gamma. There are a number of distinct advantages.

In the first place, Learning Γ saves computational resources. This is due to the fact that the input u_n lies in a lower-dimensional subspace of the high-dimensional space $X \times X$ and we can learn Γ on a small subset of the space. In practice, if the input is of a lower dimension, one may easily embed it into the space $X \times X$ by padding the vector u_n with zeroes.

Secondly, the function Γ is known to be stable. Learning Gamma makes use of equations 4.6 which in turn employs g , a function possessing the USP, and offers distinct advantages with regards to stability in the presence of perturbation. It is desirable that inputs that differ only slightly would have only ‘slight’ effects on the output of the system. Moreover, one would hope to prevent numerical errors originating from input and measurement noise (see [?, Th. 5]). As the data fed into the system is a function of both the input given and the parameters of the system itself (i.e. the value a , α in \tanh), we investigate both input-related and parameter-related stability. Both these notions are defined by way of the continuity of an encoding map of an input and design parameter respectively in [?].

Informally, we may consider input-related stability as concerned with the question of whether or not small variations in input would result in small responses. If two inputs u and u_{noisy} are close in the product topology, then their tails could differ greatly due to the metric’s insensitivity to the differences in sequence’s tails. However if the system has the USP then the function h is continuous and so $h(u), h(u_{noisy})$ remain near – maintaining a measure of robustness to input and measurement noise.[?]. In a similar fashion, parameter-related stability is obtained thanks to a result [?, Lemma 3.2] relating this form of stability with the ESP (which is here equivalent to the USP).

¹⁷ *Thirdly, Γ is globally dissipative whereas G_T is not guaranteed to be [?]. Global dissipativity prevents large numerical errors due to input data. If a system is not globally dissipative, negligibly small errors result in major errors which thus induce predictions to fail utterly. If one learns G_T directly, we are in danger of replicating the problems arising for Takens.* EdN:17

¹⁷EDNOTE: B: Would you please help me with the writing here? Otherwise I shall leave this out.

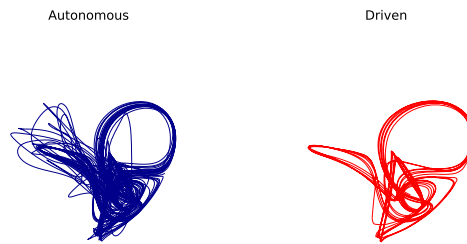


Fig. 2. Illustrating empirically the importance of global dissipativity. Plotted below the three principal components of randomly initiated trajectories of a learnt version of G_T versus the trajectory (x_{n-1}, x_n) (grey) of driven stated obtained from the Thomas Attractor's data (see below).

Chapter 5

Implementation

In this section we turn to the implementation of the function G_T by discussing a method to learn the function Γ . We then indicate where this implementation differs from previous work done by ([?]) and empirically show that the SDD of G_T are less functionally complex than $\Phi_{k,\theta}$ or T by considering the Pearson correlation coefficient. Finally, we present and discuss some numerical results for a focus system of this project: the double pendulum. We also consider other chaotic attractors.

5.1 Implementing G_T

The map G_T is implemented by applying the functions Γ , g and the projection function π_2 in the relevant order as described by the equations 4.6. To reach this point, one must first learn Γ . After first normalising the data to ensure it lies within the interval $|x| < 1$ and has zero mean value, we continue by driving the system with external input through the function g .

From (4.3) we know that $g(u, x) = (1 - a)x + a \cdot \tanh(Au + \alpha a Bx)$. Recall that g is SI-invertible.

We make use of a Recurrent Neural Network (RNN) in our implementation. RNN's are widely used in the literature for discrete-time processing and are considered to be discrete-time version of continuous dynamical systems modelling a network of neurons. (**Cite.**) A complete discussion of RNN's is outside the scope of this thesis, so we present only the details necessary to the current discussion. For information, (**Cite, Cite, Cite**) may be consulted. We implement our RNN in the *Python* programming language and making use of the *Scipy* and *Tensorflow* libraries.

If the RNN we will be using in the next step has N neurons, then A and B are each $N \times N$

matrices. Inputs u_n of dimension $K < N$ are padded by zeroes to obtain an input of dimension N . As we do not need the entire left-infinite history of input, an arbitrary initialisation of the system with initial values followed by the driving of the system with the function g means that the generated values will approach the actual solution's elements in a uniform manner thanks to the UAP. **(Cite.)**

Thirdly we prepare the network. Utilising a network N neurons (the same value N as in the dimensions of the matrices A, B). Training is done using the Adam Optimiser where the MSE loss function is optimised. The system is trained several times (in our case 4) with incrementally smaller learning rates on each run (0.001 and thereafter divided by 10 at each run). Additional parameters such as the number of network-layers, neurons per layer and training-length are presented case-by-case below.

We then reach the prediction step. Making use of g 's SI-invertibility and g as defined in (4.3), we have the following expression for u_n replicated from (4.4).

$$u_n := A^{-1} \left(\overline{\tanh}^{-1} \frac{1}{a} (x_{n+1} - (1-a)x_n) \right) - \alpha B x_n$$

The two model equations for learning from data ((4.6)) allows us to determine what the state x_{n+1} at time $n+1$ will be. We specifically remind ourselves of

$$x_{k+1} = \pi_2 \circ (\Gamma, \pi_2) \circ (\pi_2, g)(u_k, x_k)$$

5.2 Changes in this thesis/project/report to previous work of [?]

We made several changes which gave improved results:

1. Using Delay Coordinates (DC) when driving the system vs. using Single/Multi Delay Dynamics (SDD/MDD) after driving (and before training). Explain/show why these two methods are equivalent. ¹⁸

EdN:18

2. Sparse matrix implementation. Slight improvement in some scenarios, but not fantastic. ¹⁹
(Possible paragraph here/later on implementing sparsity of A/B: Generate sparse matrix that is also well-conditioned and correctly scaled matrix.)

3. Density of matrices A/B (separate from sparse matrix implementation) ²⁰

¹⁸EdNOTE: Would MDD be not computationally intensive ? So may be you just explain if you can't show it.

¹⁹EdNOTE: Did it not help you a great deal with the DP forecasting?

²⁰EdNOTE: Is it the probability distributions of the entries in A and B?

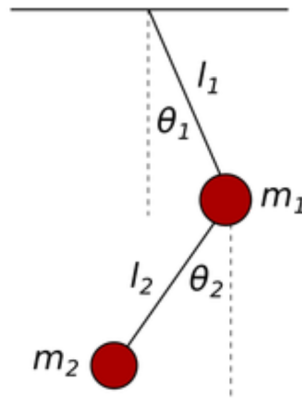
EdN:19

EdN:20

5.3 Numerical simulations

5.3.1 Double Pendulum

The Double Pendulum consists of two pendulums fastened to one another such that the system moves as a whole. The first (or higher pendulum) is attached to a fixed point while the second is attached to the endpoint of the first. (See below)



This system has four variables: angle and angular velocity of each pendulum rod. An ideal system is assumed where the rods are massless and no friction in the pivots are considered. The scalar equations of motion are derived using Newtonian physics and can be found in [?]. The second-order system is converted to a first-order ODE and solved using the Runge-Kutta numerical method.

This system is a classical example of a system exhibiting chaos. **Cite.**

- Why data dying down doesn't work (ie. Not surjective)
- Parameter Values & Success

5.3.2 Other attractors I've experimented with: Clifford & Thomas

We follow the methodology presented in [?] but opt to consider different attractors this time.

- Thomas:

$$u_x = \sin(y) - \beta x \quad (5.1)$$

$$u_y = \sin(z) - \beta y \quad (5.2)$$

$$u_z = \sin(x) - \beta z \quad (5.3)$$

- Clifford:

$$\frac{\sin(ay) + c \cos(ax)}{\sin(bx) + d \cos(by)} \quad (5.4)$$

One is easily reminded of the Lorenz system, another three-equation deterministic model which exhibits chaotic behaviour.

5.4 Generalized Pearson correlation coefficient for the DP and other attractors

Insert table: D.Pendulum, Lorenz, Others (Clifford & Thomas)

5.4. GENERALIZED PEARSON CORRELATION COEFFICIENT FOR THE DP AND OTHER ATTRACT

Input	Dimension of X	u_n vs u_{n+1}	$\begin{matrix} x_{n-1} \\ x_n \end{matrix}$ vs $\begin{matrix} x_n \\ x_{n+1} \end{matrix}$
(w_n) is the Lorenz states, sampled every 0.1 timestamps. $u_n = \frac{1}{100}w_n$ $u_n = \frac{1}{10}(\sin(0.1w_{n,x}) + \sin(0.1w_{n,y}) + \sin(0.1w_{n,z}))$	10 100 1000 10 100 1000	0.9311 0.8401	0.9731 0.9934 0.9930 0.9392 0.9616 0.9737
(w_n) comes from the Thomas Attractor (for $\beta = 0.1056$): $w_{n+1} = \begin{bmatrix} \sin(w_{n,y}) - \beta w_{n,x} \\ u_y = \sin(w_{n,z}) - \beta w_{n,y} \\ u_z = \sin(w_{n,x}) - \beta w_{n,z} \end{bmatrix}$ $u_n = w_n - \bar{w}$	10 100 1000	0.99989 	0.999886 0.999913 0.999901
(w_n) comes from Noise-Added Thomas Attractor (for $\beta = 0.1056$): $w_{n+1} = \begin{bmatrix} \sin(w_{n,y}) - \beta w_{n,x} \\ u_y = \sin(w_{n,z}) - \beta w_{n,y} \\ u_z = \sin(w_{n,x}) - \beta w_{n,z} \end{bmatrix}$ $u_n = w_n - \bar{w}$	10 100 1000	0.9994 	0.99985 0.99991 0.999895
(w_n) comes from the Thomas Attractor (for $\beta = 0.208186$): $w_{n+1} = \begin{bmatrix} \sin(w_{n,y}) - \beta w_{n,x} \\ u_y = \sin(w_{n,z}) - \beta w_{n,y} \\ u_z = \sin(w_{n,x}) - \beta w_{n,z} \end{bmatrix}$ $u_n = w_n - \bar{w}$	10 100 1000	0.99982 	0.99981 0.999851 0.99984
(w_n) comes from the Clifford Attractor (for $a = -1.7; b = 1.8; c = -1.9; d = -0.4$): $w_{n+1} = \begin{bmatrix} \sin(aw_{n,y}) + c \cos(aw_{n,x}) \\ \sin(bw_{n,x}) + d \cos(bw_{n,y}) \end{bmatrix}$ $u_n = w_n - \bar{w}$	10 100 1000	-0.21713 	0.78719 0.8334 0.84854
(w_n) comes from the Logistic map: $w_{n+1} = 4w_n(1 - w_n)$ $u_n = w_n - 0.5$	10 100 1000	-0.0314 	0.5577 0.7150 0.7826
(w_n) comes from the Svensson Attractor (for $a = -1.7; b = 1.8; c = -1.9; d = -0.4$): $w_{n+1} = \begin{bmatrix} d \sin(aw_{n,x}) - \sin(bw_{n,y}) \\ c \cos(aw_{n,x}) + \cos(bw_{n,y}) \end{bmatrix}$ $u_n = w_n - \bar{w}$	10 100 1000	-0.2243 	0.49698 0.63310 0.71151

Table 5.1. Multidimensional Correlation Coefficients ρ to indicate a general reduction in the functional complexity of the map G_T that arises from a RNN. Each row corresponds to a different dynamical system used in the experiments in [?]. In the second column corresponds to the dimension of X (number of artificial neurons) in the RNN used. The numerical estimate of ρ for the relevant vectors is plotted in the last two columns. The reader can compare the linear relationship between u_n vs $T(u_n)$ and the linear relationship $\begin{bmatrix} x_{n-1} \\ x_n \end{bmatrix}$ vs $G_T \left(\begin{bmatrix} x_{n-1} \\ x_n \end{bmatrix} \right)$. The dimension of the RNN from (??) is also given, showcasing that increasing the dimension of the driven system typically yields a map G_T of lower functional complexity.

Chapter 6

Conclusion

The driven states (x_{n-1}, x_n) holds the information of the left-infinite history of the input. Furthermore, there is some evidence that the inverse-limit space (not system) seems to distinguish two dynamical systems that are not topologically conjugate for a class of systems in the sense their inverse-limit spaces are not homeomorphic to each other. So the relation Y_T , which is homeomorphic to the inverse-limit system, may be unique to a class of systems. (The reference is a paper on Ingram's Conjecture by Marcy Barge, Henk Bruin and Sonja Stimag.) Further investigation is needed.