Theory and Applications of

Multifunctional Reservoir Computers

Andrew Flynn

NATIONAL UNIVERSITY OF IRELAND, CORK

SCHOOL OF MATHEMATICAL SCIENCES

DISCIPLINE: APPLIED MATHEMATICS

A thesis submitted for the degree of

Doctor of Philosophy

Successfully defended on 12^{th} June 2023

Head of School:	Dr. Kevin Hayes
Head of Discipline:	Prof. Sebastian Wieczorek
Supervisors:	Dr. Andreas Amann
	Dr. Vassilios A. Tsachouridis

I, Andrew Flynn, certify that the work I am submitting in this thesis is my own and has not been submitted for another degree, either at University College Cork or elsewhere. All external references and sources are clearly acknowledged and identified within this thesis. I have read and understood the regulations of University College Cork concerning plagiarism.

Andrew Flynn

Andrew Flynn

Contents

List of Works

List of publications

Some of the content in this Thesis has been included in the following papers:

- A. Flynn, V. A. Tsachouridis, and A. Amann, "Multifunctionality in a reservoir computer," *Chaos* **31**, 013125 (2021)
- A. Flynn, J. Herteux, V. A. Tsachouridis, C. Räth, and A. Amann, "Symmetry kills the square in a multifunctional reservoir computer," *Chaos* **31**, 073122 (2021)
- A. Flynn, O. Heilmann, D. Köglmayr, V. A. Tsachouridis, C. Räth, and A. Amann, "Exploring the limits of multifunctionality across different reservoir computers," *Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), IEEE* (2022)
- A. Flynn, V. A. Tsachouridis, and A. Amann, "Seeing double with a multifunctional reservoir computer," *arXiv preprint arXiv:2305.05799* (2023)
- A. Flynn, V. A. Tsachouridis, and A. Amann, "Harnessing multistability: Expanding the capabilities of neural networks," *in preparation* (2023)
- A. Flynn, C. McCafferty, F. David, V. A. Tsachouridis, V. Crunelli, and A. Amann, "Modelling seizure dynamics with a multifunctional reservoir computer," *in preparation* (2023)

- A. Flynn, V. A. Tsachouridis, and A. Amann, "Global control of multistable dynamical systems with a reservoir computer," *in preparation* (2023)
- A. Flynn, V. A. Tsachouridis, and A. Amann, "Investigating the role of untrained attractors in reservoir computers," *in preparation* (2023)
- A. Flynn, V. A. Tsachouridis, and A. Amann, "Coexisting generalised synchronisations: The backbone of multifunctional reservoir computers," *in preparation* (2023)

List of presentations

Some of the above work has also been presented at the following seminars and conferences:

- Seminar on, 'Multifunctional Neural Networks', at the Applied Mathematics Seminar Series 19/20 in University College Cork.
- Oral presentation titled, 'Multifunctionality and behind-the-scenes bifurcations in a Reservoir Computer', at Dynamic Days Digital 2020.
- Seminar on, 'Exploring Multifunctionality in a Reservoir Computer', presented to the Special Interest Group on Machine Learning & Dynamical Systems at the Alan Turing Institute, 2021.
- Oral presentation titled, 'From Seeing Double to Chaotic Itinerancy with a Multifunctional Reservoir Computer', at Dynamic Days Digital 2021.
- Poster presentation titled, 'Modelling epilepsy with a multifunctional reservoir computer' at Dynamic Days (US), Georgia Institute of Technology, 2022.
- Oral presentation titled, 'One reservoir computer to control all attractors', at the Irish SIAM Student Chapter Conference 2022, University of Limerick.

- Seminar on 'Data-driven modelling of epilepsy with a multifunctional artificial neural network', at the Dept. of Anatomy & Neuroscience Seminar Series 21/22, University College Cork.

- Invited presentation titled, 'Modelling transitions in epileptic seizure dynamics with a multifunctional reservoir computer', at the British Applied Mathematics Colloquium 2022, Loughborough University.

- Invited presentation titled, 'Dynamics and Applications of Multifunctional Reservoir Computers', at the workshop on Advances in Bio-inspired and Neuromorphic Electronics, Loughborough University, 2022.

- Contributed talk titled, 'From Seeing Double to Modelling Seizure Dynamics with Multifunctional Reservoir Computers', at the Online Conference on Differential Equations for Data Science 2023 (DEDS2023), hosted by Kanazawa University, 2023.

- Contributed presentation titled, 'Dynamics and Applications of Multifunctional Reservoir Computers', at the SIAM Conference on Applications of Dynamical Systems (DS23), Portland, Oregon, USA, 2023.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Andreas Amann, my enterprise mentor, Dr. Vassilios A. Tsachouridis, and the Irish Research Council. Thank you for giving me the opportunity to fulfil my dreams.

To Andreas, there is an endless list of things to thank you for. Above all, thank you for your patience and for believing in me. Thank you for always being so generous with your time, particularly for the numerous meetings we had that stretched long into the night. You have given me the greatest gift any student could ask for, the ability to harness my own thoughts and place them into reality. To Vassilios, thank you for also believing in me, fighting the many battles to keep the show on the road, and introducing me to the world of industry through organising my internship with Collins Aerospace.

I would also like to extend my thanks to the UCC staff, especially in the School of Mathematical Sciences, and to David O'Leary for all the help provided during my years here. A special thank you to the Head of Applied Mathematics, Prof. Sebastian Wieczorek, who has given me so many opportunities, been instrumental in my development over the years, and, out of the blue one day, introduced me to reservoir computing. Another special thank you to Dr. Andrew Keane, for examining my thesis and always showing such a huge interest in my work. Thank you also to Dr. Serhiy Yanchuk, for examining my thesis and making my viva such an enjoyable experience.

Thank you to all my collaborators, to Dr. Christoph Räth, Joschka Herteux, Oliver Heilmann, Daniel Köglmayr, Dr. Cian McCafferty, Dr. François David, Prof. Vincenzo Crunelli, and more recently Dr. Jacob Morra and Prof.

Mark Daley. The time we've spent working together has always been very enjoyable and I'm so excited to see what the future holds as we continue to tell the story of multifunctionality. Thank you also to my fellow RC practitioners that I've had the pleasure of having discussions with over the years, to Dr. Andrea Ceni, Dr. Tom Carroll, Dr. Lou Pecora, Prof. Dan Gauthier, Dr. Ling-Wei Kong, Prof. Jonathan Dawes, Dr. Adrian Wong, and many more. Thank you to all who invited me to present my work and spread the word of multifunctionality at the several events I had the pleasure of attending during my PhD.

A huge thank you to my dear friend and secondary school physics teacher, Mr. Con Hurley, who inspired me at a young age to pursue a career in science, set alight my love for physics and mathematics, and without whom I definitely would not be where I belong.

My time as a PhD student would certainly have been much less craic without the company of my fellow mathematically minded friends that I've been so fortunate to have shared Room 2.11 with. To Drs. Adrián, Chris, Chun, Eduardo, Eoin, Justin, Mateusz, Paul, Pierce, Rory, Shaui, Tony, and others, I will always remember our time together. A special thank you to Chris, for always being such a huge force of encouragement and debate, to Adrián for introducing me to the work of Žižek (things haven't been the same since), to Justin for never shying away from a trip to Callanan's, to Paul for your help with bifurcation analysis and inside knowledge on many other matters, and to Pierce, my western West Cork travel companion, I'm looking forward to many more occasions where we'll have breakfast in one city and dinner in the next. Thank you also to my fellow AM dept. colleague Dr. Hassan for always showing such great interest in my work.

To all the friends that I've made during my PhD, in particular Tressa, thank you for all your encouragement. Thank you to my many housemates for all the great times we've shared in St. John's, to Kasia for all the trips to Sicilian Delights and teaching me how to be human while I was teaching machines how to learn, to Dr. Méabh, to Dr. Lisa Maire Daly, to Dr. Saoirse, to Jamie, to Célem, to Oran, to Conor, and to basically Fahd too.

Thank you to my many close friends who have given me such an abundance of support and put up without seeing me for far too long at times during my PhD. Thank you to Ruairidh and the Nash clan, to Ryan, who I'm sure suspected how things would end up after lending me his copy of *Blade Runner*, to Baz ('Barry', 'The Edge', 'Ranger Bob', 'Denser', ...), to the world-famous 'Mr.' Barry Holland, to Conor, to Rob, to Bubby ('j a r j a r j r'), to Donnchadh ('Stoolie Mc'/'Donnaconda'), to John ('JH'), to Jeams, to James, to Crunch ('Conor'), to Rudy, to Padraig, to David, to Chris, to John Cal, to Pa ('Patrick'/'Moloney'), to Gavin, to Dan, to Al, to Kevin ('KDD'), to Niall, to Eamonn ('Gearbox'), to Alan, and to many many others.

To the Flemings of the Anchor Bar, thank you for always providing such a warm and pleasant environment to enjoy a pint of Murphy's and converse about all sort of things on a Saturday night after a week of PhDing/farming. Thank you as well to my neighbours and friends from all around Courtmacsherry and Barryroe for the support over the years.

A massive thank you to Ronan and the Coakley family, to Mark Hurley, to Peter Mc, to Michael O'Brien, to James Condon, to Mike Keohane, to Patrick O'Driscoll, to Eamonn Geary, and to everyone else who helped to keep our farm alive during my father's illness.

To Julia, thank you for always being so patient with me, for being so kind and understanding, for always showing such an incredible amount of enthusiasm about everything I do, and making things special.

There are not enough words to describe how grateful I am to my father, Kevin, who has sacrificed so much for me, educated me in ways that no university can, and shown me the meaning of true grit and determination: 'when the going gets tough, the tough get going'. To my sisters, Laura and Jen, for always believing in me, especially in times where I had no belief in myself. Thank you to my wider family; to my aunts, especially Celia, who has helped me in an immeasurable way to get to where I am today and whose success provides me with such huge inspiration and ambition to always push myself further. To my uncles, in particular, Denis, for always being there for me and providing me with plenty opportunities to relax and pick some stones in Castle Lower ('the home of the stone'), to Donal, to Mike, and to my late uncle Willie, for all of the help and whose wisdom and stories of times gone by I hope to pass on to the next generation. To my late grandmother, Nanny, and her boundless love that I will forever cherish. To my late grandparents, Patty and Peg, for their support over the years and making sure I knew all my 'times tables' inside out. To all my cousins, particularly, to my godmother, Catherine, who has been such an enormous presence in my life and the library of books that she has gifted me over the years has been instrumental in developing my sense of curiosity about the world. Thank you to all the other Wheltons, to Dr. Marie, to Brian, for all of the advice on life, to Jen, and to Caroline, I will always look up to all of you and feel so lucky to have such a strong connection to a fantastic family. To Seán, who has influenced me in countless ways throughout our time growing up together, and, who I'm sure, is the person responsible for introducing me to the concept of 'the singularity' and showing me how deep the rabbit-hole goes. To the van Brees, thank you for all of the encouragement, in particular, thank you to Paul, for freeing my mind and opening my eyes to so many things.

I could not have dreamt of achieving something like this without all of you behind me.

To all characters, far and wide.

Let ‘em all come!

– Denny Moynihan

Abstract

In the pursuit of developing artificially intelligent systems there is much to be gained from dually integrating further physiological features of biological neural networks and knowledge of dynamical systems into machine learning environments. In this Thesis such a two-armed approach is employed in order to translate ‘multifunctionality’ from biological to artificial neural networks via the reservoir computing machine learning paradigm. Multifunctionality describes the ability of a single neural network that exploits a form of multistability to perform a multitude of mutually exclusive tasks. The dynamics of multifunctional RCs are assessed across several tasks and from this many new application areas are explored which include, data-driven modelling of multistability, generating chaotic itinerancy, and reconstructing dynamical transitions present in the epileptic brain.

Chapter 1

Introduction

Advancements in machine learning often arise from a two-way street between neuroscientific observation and mathematical representation. In this Thesis, we stroll through such a street by conceptualising the neurological phenomenon of 'multifunctionality' in the context of dynamical systems in order to train an artificial neural network to perform multiple mutually exclusive tasks. The main motivation behind translating multifunctionality from biological to artificial neural networks becomes clear through the topics discussed in this Introductory Chapter.

1.1 Rise of the Machines

In modern times, the increasing power and widespread use of computers has had an immensely profound impact on human society. From the way we communicate and work, to how we enlighten and entertain ourselves, these information processing machines have changed nearly every aspect of life.

Inspired by the work of Ada Lovelace who wrote the world's first computer algorithm in 1843[1] and George Boole's 'Boolean algebra'[2], it was Alan Turing who combined these concepts during his time as a PhD student to conceive of and subsequently construct a 'universal computing machine' capable of simulating any act of mathematical deduction [3, 4]. Turing's machine triggered a technological revolution that marked the beginning of new chapter in human history known as '*The Information Age*' and laid the foundations for the

[1]This algorithm appeared in [1] as the famous 'note G'.

[2]These mathematical concepts were published in 1854 in *The Laws of Thought* [2].

computers used today.

Although there are various historical references to constructions of artificial beings with intelligence dating back to times of antiquity [3], these notions suddenly became more tangible after the invention of the first computer. This led many individuals, including Turing, to pose questions on whether these machines can think or could acquire a form of intelligence or even gain consciousness [4]. It is thoughts like these that brought together John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon in 1956 to organise a conference where a new academic research area known as 'Artificial Intelligence' (AI) was born [7].

Research in AI has come a long way since 1956, enduring many winters over the years that have been thawed by the progress made across several branches of AI, in particular, in the area of Machine Learning (ML) where the focus is on developing tools that enable computers to learn and perform tasks requiring a certain level of intelligence. Today, it is through these ML tools that AI is recognised as an umbrella term referring to a suite of technologies that mimic and outperform humans on a growing list of tasks.

1.2 Two-way street

Throughout the history of ML, the brain, and insight into its inner workings, has been a constant source of inspiration in the design of new ways in which a machine can learn. Numerous mathematical models of how the brain processes information have provided the conceptual foundations needed to build a bridge from the brain to new ML paradigms. It is through this 'two-way street' between Neuroscience and Mathematics that several key advancements in AI and ML technologies have emerged from.

[3]The earliest known example of these was Talos, the giant automaton (a moving mechanical device made in imitation of a human being) forged out of brass by Hephaestus, as documented in Apollodorus' compendium of Greek myths, *Bibliotheca* [5].

[4]Turing played an important role again in consolidating these ideas and proposed the 'The Imitation Game' (also known as 'The Turing Test') in [6] as a means to measure a machine's level of intelligence.

At the core of many ML methods is an 'artificial neural network' (ANN) which is a mathematical abstraction of the basic building blocks of biological neural networks (BNNs) that the brain relies on for computation. ANNs typically consist of various interconnected artificial neurons that mimic how biological neurons process information and, for the purposes of ML, several techniques have been developed to train ANNs to perform a given task.

The study of ANNs began in 1943 with Warren McCulloch and Walter Pitts who, inspired by how the brain performs calculations using many cells that are connected together, constructed a network of artificial neurons based on threshold activation functions connected by directed weighted paths [8]. While this ANN was capable of performing only simple logic operations it set the stage for researchers to develop ANNs which, on one hand, more accurately modelled dynamical processes in the brain, and on the other hand, could be exploited for computational purposes.

The recent explosion of interest in ML-based technologies is largely due to the enormous progress made in the development of new techniques for training ANNs to solve more and more sophisticated problems. From aiding professionals in healthcare [9, 10] and agriculture [11–13], to advances in autonomous vehicles [14–16], and beating world champions in the game Go [17], various designs of ANNs have quickly become a dominating factor in driving forward technological innovation.

While it is fair to say that the rapid increase in ML applications can also be attributed to the wider accessibility of computers with sufficient processing power to train large ANNs, the two-way street analogy has been central to many of the developments mentioned above. For instance, the famous AlexNet [18] and Inception [19] network architectures explored in [11] on machine vision problems in precision farming are based on 'convolutional neural networks' (CNNs). CNNs are a type of ANN inspired by Hubel and Wiesel's insights into the structure of the visual cortex in a cat's brain [20] that were realised by making use of the convolution operation in functional analysis as introduced by LeCun *et al.* in [21].

It is clear from the success of these technologies that there is much to be gained in the pursuit of developing artificially intelligent systems by articulating our current knowledge of the brain and mathematics in ML environments. In this Thesis, we employ this bilateral rationale to broaden the computational capabilities of certain ANNs with inspiration from 'multifunctional neural networks'.

1.3 Harnessing Multifunctionality

Multifunctional neural networks are networks of biological neurons that are capable of performing more than one task without changing any network connections [22]. Multifunctionality is a well observed phenomenological feature of BNNs found in the brain of humans and many other creatures [23–27].

In [22], Briggman and Kristan Jr. argue that the presence of multifunctionality reflects the developmental history of the brain and that some multifunctional neural networks may have evolved from networks that initially performed only one task. As multifunctionality is typically found in networks with a relatively small number of neurons, it is thought that multifunctionality may naturally emerge from an efficient use of limited resources, in this case neurons, in order for a given species to adapt to environmental changes.

It is natural to consider that many more network systems may have the ability to achieve multifunctionality. Where this becomes immediately relevant is in the domain of ML. If multifunctionality can be harnessed from a ML perspective, it can be used to unlock additional computational capabilities of certain ANNs which would otherwise have remained dormant.

Moreover, training ANNs to achieve multifunctionality is highly advantageous from a practicality point of view as it has the potential to broaden the networks functional capacity to solve multiple tasks using the same set of trained weights. Many more advantages of multifunctionality and the doors it opens to new ML application areas are discussed in Sec. 3.4.

Taking all of the above into consideration, the aim of this Thesis is to investigate how multifunctionality can be translated as a phenomenon of BNNs to ANNs using the 'reservoir computing' [28–30] approach to ML. In brief, a 'reservoir computer' (RC) is a dynamical system that can be realised as an ANN and an overview of this ML paradigm is provided in Chapter 2.

As outlined in Sec. 3.1, the main breakthrough in this Thesis comes from interpreting the characteristic features of multifunctionality through the lens of dynamical systems. The critical insight that is needed to build the bridge from the brain to design tools which enable RCs to achieve multifunctionality comes by realising that BNNs exploit a form of multistability in order to exhibit multifunctionality. From this, the concept of 'generalised synchronisation', which RCs rely on in order to learn, is extended to describe how multifunctionality can be achieved in a RC in Sec. 3.2.1. In Sec. 3.2.2, a new training technique is developed that enables RCs to perform tasks of a multifunctional nature throughout this Thesis.

In the Chapters that follow, some of the extent and limitations of what can be achieved with a multifunctional RC are explored through several numerical experiments. These experiments reveal key relationships between multifunctionality and RC training parameters. This subsequently provides the foundational knowledge required to develop application areas for multifunctionality and paves the way for data-driven modelling of either known or suspected multistabilities present in experimental real-world data.

Throughout these experiments and applications of multifunctionality a key emphasis is placed on understanding how a RC learns to achieve multifunctionality. In certain circumstances, the particular bifurcations the RC goes through that are responsible for enabling learning, preventing learning, and producing unintended functionalities are identified.

A more detailed breakdown on these contributions is provided in the following section.

1.4 Thesis Layout

The contributions made in each Chapter of this Thesis are outlined as follows:

- Chapter 2 highlights some of the main concepts and advancements made in reservoir computing over the years. A well-known task is used to illustrate how to train a RC to reconstruct the dynamics of a chaotic attractor. It is by exploring the dynamics of the trained RC from a new perspective that fresh ground is broken on establishing how a RC learns, these surprising results serve as additional motivation towards pursuing the main objectives of this Thesis.

- Taking inspiration from examples of multifunctionality in the brain of humans and other living creatures, the focus of Chapter 3 is the translation of multifunctionality from BNNs to ANNs. The steps involved in training a RC to achieve multifunctionality are introduced and the unique abilities and advantages of multifunctional RCs are compared with several other related ML approaches.

- Multifunctionality is first realised in a RC in Chapter 4 and the results presented here have been published in [31].

- The issues related to symmetry and multifunctionality are considered in Chapter 5. Much of the results presented in this Chapter are published in [32]. Elements of discussions regarding symmetry and RCs are adapted from [32] and were contributed by Joschka Herteux with input from Christoph Räth.

- In Chapter 6 the issue of overlapping training data is explored across different RCs. The work regarding the 'leaky integrator' RC and 'next generation' RC was carried out by Oliver Heilmann and Daniel Köglmayr with input from Christoph Räth and appeared in [33]. The relationship between multifunctionality and overlapping training data is then explored with the 'continuous-time' RC in a paradigmatic scenario using the 'seeing double' problem. The insight gained from the results presented here led towards establishing the conceptual backbone behind how multifunctionality is achieved in a RC in Chapter 3. By investigating the dynamics of the RC as it solves the seeing double problem the bifurcations responsible for the rise and fall of multifunctionality are identified. This work is presently under review, for a preprint see [34].

- Chapter 7 continues on from the research in Chapter 6 by reinterpreting an extreme case of the seeing double problem as a benchmark task to test the limits of multifunctionality across different RCs. The results presented in this Chapter are available in [33] and the work regarding the 'leaky integrator' RC and 'next generation' RC was carried out by Oliver Heilmann and Daniel Köglmayr with input from Christoph Räth.

- The research presented in Chapter 8 showcases the first application of multifunctional RCs as the techniques developed in this Thesis are used to model seizure dynamics. This work is from of an ongoing collaboration with Cian McCafferty, François David, and Vincenzo Crunelli who provided the experimental data and helped in framing the goals of the numerical experiments carried out here.

- Throughout all of the research mentioned above that is either published or still in preparation, the author of this Thesis is the first author on each paper and also instigated and lead the collaborative work. Vassilios A. Tsachouridis and Andreas Amann are co-authors in each of these works.

Chapter 2

Reservoir Computing

Presented in this Chapter is an overview of reservoir computing, an outline of the main type of RC that is studied throughout this Thesis, and an example of training this RC in time-series prediction problems. An analysis of the RC's dynamics after training provides new insight behind how this machine learns and reveals the existence of 'untrained attractors' which serves as further motivation towards addressing the main questions explored in this Thesis regarding multifunctionality.

2.1 Overview

The reservoir computing approach to ML received its name from Verstraeten *et al.* in [28] who coined the term reservoir computer as a means to unify the shared philosophies of Echo-State Networks [29] (ESNs) and Liquid-State Machines [30] (LSMs). These are two independently proposed designs of ANNs with recurrent connections (RNNs) that were introduced in the early 2000s that have since become increasingly synonymous under the term RC and are the pillars upon which this ML paradigm has been built.

2.1.1 RC Philosophy

While Jaeger's ESNs were engineered in the context of machine learning and Maass's LSMs were developed from a computational neuroscience perspective, they share a similar philosophy. Instead of training all the weights in a network it is sufficient to optimise only the weights of a readout layer to solve a given task. This innovation, which distinguishes RCs amongst other ML approaches, arose as a means to avoid several issues associated with training RNNs via back-propagation through time [29]. When comparing the performance of RCs against other ML approaches across different benchmark tasks, the common conclusion is that RCs can outperform other ML methods with significantly less computational expense [35–37].

This ideological shift in training RNNs stems from the design of a suitable internal layer, known as the 'reservoir', which does not need to be trained according to a particular problem. Instead, so long as the reservoir enables the state of the RC to become a representation of the history of driving inputs then only a suitable readout layer needs to be found in order to project this information out of the RC.

2.1.2 Opening the Black-Box

Another driving factor behind the popularity of reservoir computing in both academia and industry is that a RC can be realised as both an ANN and a dynamical system. By describing an artificially intelligent system as a dynamical system then the same set of mathematical tools, that have dramatically improved our understanding of the world around us, can in principal be used to explain how these machines learn.

The success of examining the behaviour of RCs from a dynamical systems perspective is well documented in the literature with many key advances that have improved our knowledge of how RCs learn such as those reported in [38–42]. The results presented in this Thesis compliment these by establishing the particular bifurcations a RC goes through in order to achieve multifunctionality in certain circumstances.

These results shed some light on how RCs learn and perform certain tasks and these efforts relate to the problems of 'explainability' and 'opening the black-box' that the much wider AI research community are focused on addressing [43–45]. The focus of 'Explainable AI' is to develop an understanding of how the machine performs certain tasks at the level where a human can understand decisions made by the machine. The aim of 'opening the black-box' is to tackle the Achilles' heel of many ML methods that are considered to be a black-box, a system where there is nothing known about how the system produces a particular output for a given input. This results in little being known about the internal mechanisms responsible for enabling learning, preventing learning, or producing unintended biases or functionalities. The growing need to address the widening gap between explainability and highly disruptive innovations in ML and the problems this may produce for humanity are discussed in Appendix A.

Furthermore, through this lens of dynamical systems it is also highly feasible to translate further behavioural features of BNNs to an artificial setting and it is through this point of view that the realisation of multifunctionality in a RC is achieved. To comment further, biological based learning rules have also been developed for RCs by following this philosophy [46,47]. Biologically-inspired RCs have also been created by integrating certain aspects of BNN topologies into ANN designs such as the fruit fly connectome derived RC in [48–50].

2.1.3 Application Areas

As interest in RCs has continued to grow over the years, many application areas have been developed. RCs have been trained to perform a large variety of tasks, for example, decoding human emotions [51] and neuronal information in brain-machine interfaces [52], model-free control of nonlinear systems [53, 54], predicting critical transitions [55, 56], assisting in the diagnosis of Parkinson's disease [57], reconstruction and prediction of trajectories on chaotic attractors [58, 59], visual identification tasks [60, 61], real-time detection of epileptic seizures [62], and inferring from limited time series data; unmeasured state variables [63], Lyapunov exponents [64], and causal dependencies between variables [65].

By training RCs to become multifunctional, this brings multistability into the world of ML and opens the door for many new applications areas.

2.2 Principle of Operation

The principle of operation shared amongst most RC setups, in particular for time series prediction and attractor reconstruction tasks, is described as follows. This description is adapted from Lu *et al.* in [59].

2.2.1 Using a RC

Task description

Consider an autonomous continuous-time dynamical system that evolves according to,

$$\dot{\boldsymbol{x}}(t) = \mathbf{g}\left(\boldsymbol{x}(t)\right), \tag{2.1}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{2.2}$$

where, $\boldsymbol{x}(t) \in \mathbb{R}^K$ is the state of the system at a given time t, $\mathbf{g}(\cdot) : \mathbb{R}^K \to \mathbb{R}^K$ describes the governing equations of the system which may be unknown, and $\boldsymbol{x}_0 \in B_{\mathcal{Z}} \subset \mathbb{R}^K$ where $B_{\mathcal{Z}}$ is the basin of attraction of some attractor $\mathcal{Z}$. An attractor, $\mathcal{Z}$, is defined as a closed, invariant, and minimal set that attracts all $\boldsymbol{x}_0 \in B_{\mathcal{Z}}$ such that the distance between $\boldsymbol{x}(t)$ and $\mathcal{Z}$ tends to 0 as $t \to \infty$ [66]. Consider a vector $\boldsymbol{u}(t) \in \mathbb{R}^D$ that describes the measurement of D variables of $\boldsymbol{x}(t)$ at time t. The relationship between $\boldsymbol{u}(t)$ and $\boldsymbol{x}(t)$ is given by,

$$\boldsymbol{u}(t) = \mathbf{h}\left(\boldsymbol{x}(t)\right), \tag{2.3}$$

where $\mathbf{h}(\cdot) : \mathbb{R}^K \to \mathbb{R}^D$ is the 'measurement function' that maps $\boldsymbol{x}(t)$ to $\boldsymbol{u}(t)$ at each time t. The aim is to train a RC to predict future values of $\boldsymbol{u}(t)$ beyond some time $t = t_{\text{train}}$.

While it is possible that $D \ll K$, in most experiments carried out in this Thesis $D = K$ and $\mathbf{h}(\cdot)$ is the identity matrix of the appropriate size, so $\boldsymbol{u}(t) = \boldsymbol{x}(t)$.

Open-loop training

There are two stages involved in training the RC, a listening stage and a training stage. In both stages the state of the RC evolves according to,

$$\dot{\boldsymbol{r}}(t) = \boldsymbol{f}\left(\boldsymbol{r}(t), \boldsymbol{u}(t)\right), \tag{2.4}$$

$$\boldsymbol{r}(0) = \boldsymbol{r}_0, \tag{2.5}$$

where $\boldsymbol{r}(t) \in \mathbb{R}^N$ is the state of the RC at time t, $\boldsymbol{f}(\cdot, \cdot) : \mathbb{R}^N \times \mathbb{R}^D \rightarrow \mathbb{R}^N$ describes the RC formulation and in general $N \gg D$. $\boldsymbol{r}_0$ is typically chosen as $\mathbf{0}^T = (0, \ldots, 0)^T$ where T in this instance denotes the transpose operation. This nonautonomous dynamical system is referred to as the 'listening RC' where $\boldsymbol{u}(t)$ is used as input in order to drive a response from the state of the RC from $t = 0$ to $t = t_{\text{train}}$.

The listening stage is described as the duration of time where the listening RC is driven by $\boldsymbol{u}(t)$ for $0 \leq t \leq t_{\text{listen}}$ where $t_{\text{listen}} < t_{\text{train}}$ is chosen such that $\boldsymbol{r}(t)$ is determined by a history of driving inputs and is no longer dependent on its initial condition, $\boldsymbol{r}_0$. The sequence of events that give rise to this phenomenon are discussed in Sec. 2.2.2.

The training stage is described as the duration of time where the listening RC is driven by $\boldsymbol{u}(t)$ for $t_{\text{listen}} \leq t \leq t_{\text{train}}$. Training consists of finding a function $\hat{\boldsymbol{\psi}}(\cdot) : \mathbb{R}^N \rightarrow \mathbb{R}^D$ that minimises,

$$\int_{t=t_{\text{listen}}}^{t_{\text{train}}} \frac{1}{t_{\text{train}} - t_{\text{listen}}} ||\hat{\boldsymbol{\psi}}\left(\boldsymbol{r}(t)\right) - \boldsymbol{u}(t)||_2^2 \, dt, \tag{2.6}$$

where $\hat{\boldsymbol{\psi}}(\cdot)$ is called the 'readout function' and can, for instance, be computed using a regression technique like the method outlined in Sec. 2.3.2.

In order for a RC to achieve multifunctionality, the properties required for the pair of functions $(\boldsymbol{f}, \hat{\boldsymbol{\psi}})$ are discussed in Sec. 3.2.1 and elucidated through several numerical experiments during Chapters 4 - 8.

Closed-loop predicting

After the training, $\boldsymbol{u}(t)$ is replaced by $\hat{\boldsymbol{\psi}}\left(\boldsymbol{r}(t)\right)$ in Eq. (2.4) and the evolution of the state of the 'predicting RC' is described by the following closed-loop system,

$$\dot{\hat{\boldsymbol{r}}}(t) = \boldsymbol{f}\left(\hat{\boldsymbol{r}}(t), \hat{\boldsymbol{\psi}}\left(\hat{\boldsymbol{r}}(t)\right)\right), \tag{2.7}$$

$$\hat{\boldsymbol{r}}(0) = \boldsymbol{r}(t_{\text{train}}). \tag{2.8}$$

While $\hat{\boldsymbol{r}}(t)$ and $\boldsymbol{r}(t)$ are both $\in \mathbb{R}^N$, to distinguish the dynamics of $\hat{\boldsymbol{r}}(t)$ from $\boldsymbol{r}(t)$, $\hat{\boldsymbol{r}}(t)$ is defined as $\hat{\boldsymbol{r}}(t) \in \mathbb{S}$ where $\mathbb{S}$ is referred to as the 'RC's state space' and is used henceforth when discussing the dynamics of the closed-loop predicting RC.

By computing solutions of the above autonomous dynamical system, predictions of $\boldsymbol{u}(t)$ for $t > t_{\text{train}}$, denoted as $\hat{\boldsymbol{u}}(t)$, are given by,

$$\hat{\boldsymbol{u}}(t) = \hat{\boldsymbol{\psi}}\left(\hat{\boldsymbol{r}}(t)\right). \tag{2.9}$$

Again, while both $\boldsymbol{u}(t)$ and $\hat{\boldsymbol{u}}(t)$ are $\in \mathbb{R}^D$, to distinguish the dynamics of $\hat{\boldsymbol{u}}(t)$ from $\boldsymbol{u}(t)$, $\hat{\boldsymbol{u}}(t)$ is defined as $\hat{\boldsymbol{u}}(t) \in \mathbb{P}$ where $\mathbb{P}$ is referred to as the 'prediction state space' and is used henceforth when discussing the dynamics of the closed-loop predicting RC when projected via $\hat{\boldsymbol{\psi}}$.

In the case where $\boldsymbol{u}(t)$ describes a trajectory on a chaotic attractor, $\mathcal{A} \subset \mathbb{R}^D$, then no matter how perfect the fit that $\hat{\boldsymbol{\psi}}$ provides, $\hat{\boldsymbol{u}}(t)$ will diverge from $\boldsymbol{u}(t)$ after some finite time $t = t' > t_{\text{train}}$. As a result the predicting RC will only ever be capable of providing reasonable short term predictions of $\boldsymbol{u}(t)$, according to a chosen error criteria such as $||\hat{\boldsymbol{u}}(t) - \boldsymbol{u}(t)||_2$ being close to 0, for $t < t'$ where t' is the time where $||\hat{\boldsymbol{u}}(t) - \boldsymbol{u}(t)||_2$ exceeds a chosen threshold.

However, even though $\hat{\boldsymbol{u}}(t) \neq \boldsymbol{u}(t)\ \forall\ t > t'$, if the predicting RC has achieved 'attractor reconstruction' then the long term dynamical characteristics of $\hat{\boldsymbol{u}}(t)$ will be indistinguishable from $\boldsymbol{u}(t)$. By resembling the long term dynamics it is meant here and in later uses of the expression that, for instance, the dynamics of the attractors described by $\boldsymbol{u}(t)$ and $\hat{\boldsymbol{u}}(t)$ will have nearly identical Poincaré sections when computed for the same region of $\mathbb{R}^D$ and $\mathbb{P}$ as $t \to \infty$ for $t > t_{\text{train}}$.

When attractor reconstruction occurs we say that there exists an attractor $\mathcal{S} \subset \mathbb{S}$ such that when the state of the predicting RC approaches $\mathcal{S}$ and is projected from $\mathbb{S}$ to $\mathbb{P}$ via $\hat{\boldsymbol{\psi}}\left(\cdot\right)$, the dynamics of the 'reconstructed attractor',

$\hat{\mathcal{A}} \subset \mathbb{P}$, resembles the dynamics of $\mathcal{A}$ in the long term, i.e., $\mathcal{A}$ and $\hat{\mathcal{A}}$ will, as mentioned above, have nearly identical Poincaré sections. A more precise definition of what it means for a RC to achieve attractor reconstruction is provided in the following Sec. 2.2.2.

2.2.2 Perspectives on Learning

There are a number of concepts that try to establish the particular mechanisms of how $\boldsymbol{r}(t)$ becomes dependent on a history of driving inputs from $\boldsymbol{u}(t)$ during the listening stage and how attractor reconstruction is achieved. An outline on some of these concepts are provided below.

Echo State Property

In [29], Jaeger put forward the 'echo state property' (ESP) as a means to describe when it is possible for an ESN, which is a RC in the form of a RNN, to be trained to reliably perform a given task. Jaeger proposes that, under strict design criteria for an ESN and if t_{listen} is large enough then, at some time t_{ESP}, where $0 \leq t_{\text{ESP}} \leq t_{\text{listen}}$, the ESN possesses the ESP. The ESP describes the scenario where the state of the ESN 'forgets' its initial condition and, for $t > t_{\text{ESP}}$, $\boldsymbol{r}(t)$ is dependent on a history of driving inputs from $\boldsymbol{u}(t)$.

On the other hand, if $\boldsymbol{u}(t)$ is removed from the listening RC for $t > t_{\text{ESP}}$ then the state of the RC will converge to an original equilibrium solution of the system. Furthermore, if the RC is designed according to the criteria specified by Jaeger in [29] then the state of the RC will approach the origin.

Generalised Synchronisation

Building on the insight that the ESP provides, in [59] Lu *et al.* put forward the concept of 'generalised synchronisation' (GS) as a means to describe the learning process for the RC and specify a set of conditions in order for the RC to provide accurate short-term predictions of $\boldsymbol{u}(t)$ and reconstruct the long-term dynamics of the attractor it describes.

Synchronisation occurs where two or more coupled nonidentical systems evolve together in time such that the states of each respective system become correlated or even identical. The multiple different ways in which dynamical systems can synchronise are detailed in books such as [67] and [68].

In [69], Pecora and Carroll made a significant breakthrough by demonstrating that synchronisation can also occur even if the respective system's dynamics are chaotic. This result led towards Rulkov *et al.* introducing the concept of GS in [70] to describe the scenario where two or more coupled nonidentical systems can become synchronised even though the evolution of these systems may be quite different. In [71], Kocarev and Parlitz describe GS specifically in the context of unidirectionally coupled systems, these are drive-response systems that can be written in the form of the response listening RC system described in Eq. (2.4) and driving system in Eq. (2.1). In both [70] and [71] the concept of a 'functional relation' is developed to describe GS through the established dependency between the dynamics of the coupled systems whereby the dynamics of the response system can be controlled or influenced by the driving system.

Applying the formalism of GS developed in [70] and [71] to the case of the listening RC that is driven by Eq. (2.1), the following definition clarifies what it means for GS to occur in the case of the driven listening RC.

Definition 2.2.1 (Generalised Synchronisation for a listening RC)**.** *The unidirectionally coupled system, defined by Eqs.* (2.1) *and* (2.4) *possesses the property of GS between* $\boldsymbol{x}(t)$ *and* $\boldsymbol{r}(t)$ *if* $\exists$ *a transformation,* $\boldsymbol{\phi} : \mathbb{R}^K \to \mathbb{R}^N$*, a manifold* $\mathcal{M} = \{(\boldsymbol{x}(t), \boldsymbol{r}(t)) : \boldsymbol{r}(t) = \boldsymbol{\phi}(\boldsymbol{x}(t))\}$*, and a subset* $B = B_{\boldsymbol{x}} \times B_{\boldsymbol{r}} \subset \mathbb{R}^K \times \mathbb{R}^N$ *where* $\mathcal{M} \subset B$ *and* B *is the basin of attraction where trajectories of the unidirectionally coupled system approach* $\mathcal{M}$ *as* $t \to \infty$ $\forall$ $(\boldsymbol{x}_0, \boldsymbol{r}_0) \in B$*.*

Proofs on the existence of such a $\boldsymbol{\phi}$ for a general class of unidirectionally coupled systems are outlined in [70] and [71]. However, it is Pyragas' comments on these proofs in [72] that provides an immediate connection between $\boldsymbol{\phi}$ and the ESP. Pyragas says, in the context of determining whether a given unidirectionally coupled system (defined in a similar way to Eqs. (2.1) and (2.4)) exhibits GS, there exists such a $\boldsymbol{\phi}$ "if, under the action of driving perturbations, the response system 'forgets' its initial condition". This aspect of the response system forgetting its initial condition is precisely the same as what Jaeger calls the ESP.

The 'auxiliary system approach' introduced by Abarbanel *et al.* in [73] is a method that is used to test whether GS occurs. This involves driving a given response system and an identical 'auxiliary' system with the same driving input but both response systems are initialised with different initial conditions from the same basin of attraction. If the response from the response system and the corresponding auxiliary system synchronise, then GS is said to occur. Recently, in [74] Platt *et al.* showcased the success of the auxiliary system approach when applied to RCs across different attractor reconstruction tasks where the method is used to provide a prediction on when attractor reconstruction can occur.

Achieving attractor reconstruction

As Platt *et al.* highlight in [74], the occurrence of GS does not guarantee that attractor reconstruction can be achieved. It is only when the predicting RC remains stable near the set $\boldsymbol{\phi}(\mathcal{A}) \subset \mathbb{R}^N$ that attractor reconstruction is possible. This set $\boldsymbol{\phi}(\mathcal{A})$ is defined as the image of the driving inputs during the training stage.

In [59], Lu *et al.* make use of 'transversal Lyapunov exponents' to determine the suitability of the particular readout function, $\hat{\psi}$, in reconstructing a given attractor, $\mathcal{A}$. In the case of unidirectionally coupled systems, like the drive-response system defined by Eqs. (2.1) and (2.4), the transversal Lyapunov exponents are the Lyapunov exponents of the response system under the constraint that these Lyapunov exponents are calculated for a trajectory of the response system under the influence of the drive. These Lyapunov exponents receive the name transversal as they correspond with assessing the behaviour of infinitesimal perturbations in directions which are transverse to, in this case, $\boldsymbol{\phi}(\mathcal{A})$. Lu *et al.* demonstrate that $\boldsymbol{\phi}(\mathcal{A})$ is attracting if all transversal Lyapunov exponents of the predicting RC on $\boldsymbol{\phi}(\mathcal{A})$ are negative. On the other hand, if there is any positive transversal Lyapunov exponents, then $\boldsymbol{\phi}(\mathcal{A})$ is not attracting. Lu *et al.* compute the transversal Lyapunov exponents of the predicting RC using data from the listening RC's response to the driving input signal during the training, where it is assumed GS occurs and the state of the listening RC approaches $\boldsymbol{\phi}(\mathcal{A})$. These results show precisely when there exists a stable attractor in the predicting RC's state space that corresponds to $\boldsymbol{\phi}(\mathcal{A})$.

Taking all of the above into consideration, the following set of definitions and remark more clearly describe what it means for a RC to achieve attractor reconstruction based on the concept of GS and the generation of the readout function, $\hat{\boldsymbol{\psi}}$, via the training technique.

Definition 2.2.2 (The Response Attractor)**.** $\exists$ *a stable 'response attractor',* $\mathcal{S} \subset \mathbb{S}$ *that is associated with* $\boldsymbol{\phi}(\mathcal{A})$ *if all transversal Lyapunov exponents of the predicting RC on* $\boldsymbol{\phi}(\mathcal{A})$ *are negative.*

Definition 2.2.3 (Attractor Reconstruction with a RC)**.** *Attractor reconstruction occurs if the response attractor,* $\mathcal{S}$*, created by* $\boldsymbol{\phi}(\mathcal{A})$ *and* $\hat{\boldsymbol{\psi}}$ *through GS and the training stage, exists, is stable, and when the state of the predicting RC on* $\mathcal{S}$ *is projected from* $\mathbb{S}$ *to* $\mathbb{P}$ *using* $\hat{\boldsymbol{\psi}}$*, then the long-term dynamics of the reconstructed attractor,* $\hat{\mathcal{A}}$*, resembles the long-term dynamics of* $\mathcal{A}$*, i.e.,* $\mathcal{A}$ *and* $\hat{\mathcal{A}}$ *will have nearly identical Poincaré sections when computed for the same portion of* $\mathbb{R}^D$ *as* $t \to \infty$ *for* $t > t_{train}$*.*

Remark 2.2.1 (The role of $\boldsymbol{\phi}$)**.** *If attractor reconstruction occurs, the transformation,* $\boldsymbol{\phi}$*, introduced in Definition 2.2.1, is then referred to as the 'lifting operator' as it effectively lifts* $\mathcal{A}$ *to a new attractor* $\mathcal{S}$ *in* $\mathbb{S}$*.*

The above definitions and remarks are later extended in Sec. 3.2.1 to cater for the case of multifunctionality where **a single function**, $\hat{\boldsymbol{\psi}}$, is used to project the dynamics of **multiple coexisting attractors** from $\mathbb{S}$ to $\mathbb{P}$.

Returning to where the story of synchronisation in chaotic system began, the reservoir computing approach to ML represents a realisation of Pecora and Carroll's intuitions as outlined in the closing statements made in [69][1]. Pecora and Carroll point out that by designing systems to synchronise in

[1]This statement comes from comments made by Lou Pecora during a seminar he gave at University College Cork in September 2022.

particular ways, this may open many new applications areas for dynamical systems and that the process of synchronisation "could supplant the more 'fixed-point' view of neural networks" which was held at that time. In the present context of reservoir computing, the means to design a readout function is what enables the GS that occurs between a listening RC and its driving system to be exploited for many different ML applications.

2.3 RC Formulation and Training Procedure

The particular RC that is studied throughout the majority of this Thesis was presented by Lu *et al.* in [59] which is an ESN described according to the same open-loop format as in Eqs. (2.4). In this setup, the listening RC is driven by an input signal, $\boldsymbol{u}(t) \in \mathbb{R}^D$, for $0 \leq t \leq t_{\text{train}}$ and its response to this driving signal is found by generating solutions of the following nonautonomous dynamical system,

$$\dot{\boldsymbol{r}}(t) = \gamma \left[-\boldsymbol{r}(t) + \tanh\left(\mathbf{M}\,\boldsymbol{r}(t) + \sigma \mathbf{W}_{in}\,\boldsymbol{u}(t)\right)\right], \tag{2.10}$$

$$\boldsymbol{r}\,(0) = \mathbf{0}^T. \tag{2.11}$$

In Eq. (2.10), $\boldsymbol{r}(t) \in \mathbb{R}^N$ describes the state of the listening RC at a given time t and N is the number of artificial neurons in the network. γ is a decay-rate parameter arising from the derivation of this RC from the initial ESN proposed by Jaeger in [29]. The tanh 'activation function' is a pointwise operation and is defined as $\tanh(\cdot) : \mathbb{R}^N \to \mathbb{R}^N$. The topology of the network is described by the adjacency matrix, $\mathbf{M} \in \mathbb{R}^{N \times N}$. The input strength parameter, σ, and the input matrix, $\mathbf{W}_{in} \in \mathbb{R}^{N \times D}$, when multiplied together represent the weight given to $\boldsymbol{u}(t)$, as it is projected into the listening RC. Solutions of Eq. (2.10) are computed using the 4^{th} order Runge-Kutta method with time step τ. An illustration of this open-loop RC setup is shown in Fig. 2.1.

A description of how this RC is trained to reconstruct the dynamics of an attractor, $\mathcal{A} \subset \mathbb{R}^D$, described by $\boldsymbol{u}(t)$ according to the listening and training stages described in Sec. 2.2.1 is outlined as follows.

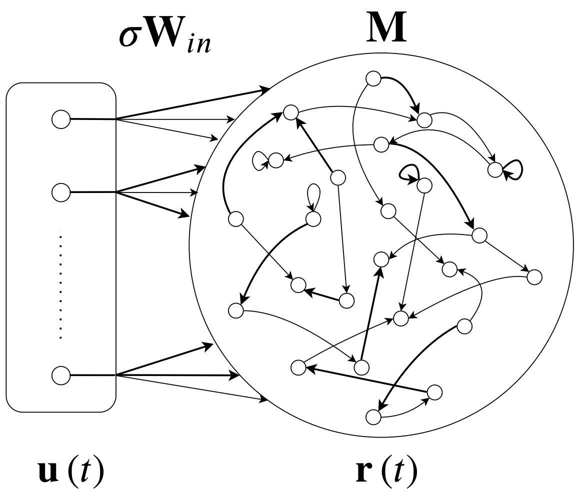


Figure 2.1: Open-loop RC: The input signal $\boldsymbol{u}(t)$ is projected by $\sigma\mathbf{W}_{in}$ to drive a response from the RC state, $\boldsymbol{r}(t)$.

2.3.1 Listening stage

In the listening stage, $\boldsymbol{u}(t)$ is used to drive the listening RC in Eq. (2.10) away from its initial condition. Unless otherwise specified, Eq. (2.10) is initialised with $\boldsymbol{r}(0)$ as described in Eq. (2.11) where $\mathbf{0}^T = (0, 0, \ldots, 0)^T$ and T denotes the transpose operation.

As outlined in Sec. 2.2.1, solutions of Eq. (2.10) are computed up to $t = t_{\text{listen}}$ where t_{listen} is chosen to be large enough such that for $t < t_{\text{listen}}$, the RC exhibits the ESP and generalised synchronisation (GS) occurs between the RC and the dynamical system where $\boldsymbol{\mathcal{A}}$ originates from.

2.3.2 Training stage

There are many ways to design $\boldsymbol{\psi}(\cdot)$ for a given task. In most examples $\boldsymbol{\psi}(\cdot)$ is given by a 'readout matrix', $\mathbf{W}_{out} \in \mathbb{R}^{D\times N}$. A brief history on the different designs of $\mathbf{W}_{out}$ that have been adopted over the years and the particular technique that is used to compute $\mathbf{W}_{out}$ for the numerical experiments carried out in this Thesis is outlined as follows.

Readout matrix design[2]

There are many ways to design $\mathbf{W}_{out}$ with respect to a given problem. While originally it was mostly linear terms of $\boldsymbol{r}(t)$ that were used to generate $\mathbf{W}_{out}$, in recent years the inclusion of quadratic terms of $\boldsymbol{r}(t)$ in the training to produce a 'square readout matrix' has become popular after its introduction by Lu *et al.* in [63], where it was employed to mitigate the influence of symmetries in the Lorenz system [75]. This 'squaring technique' was empirically found to improve a RC's performance in more general cases [35, 76]. In [77], Herteux and Räth show that this can in part be attributed to the ability of the square readout matrix to break harmful symmetries present in the equations describing a RC, i.e., when $\boldsymbol{f}\left(-\hat{\boldsymbol{r}}(t), \hat{\boldsymbol{\psi}}\left(-\hat{\boldsymbol{r}}(t)\right)\right) = -\boldsymbol{f}\left(\hat{\boldsymbol{r}}(t), \hat{\boldsymbol{\psi}}\left(\hat{\boldsymbol{r}}(t)\right)\right)$. In the context of attractor reconstruction, these harmful symmetries must be broken in order to prevent the creation of a 'mirror-attractor', an inverted version of the attractor that the RC was specifically trained to reconstruct.

The relationship between multifunctionality and symmetry in relation to mirror-attractors and the readout matrix is studied extensively in Chapter 5.

Squaring technique

Taking the above into account, for the most part of this Thesis, $\hat{\boldsymbol{\psi}}\left(\cdot\right)$ is written as,

$$\hat{\boldsymbol{\psi}}\left(\hat{\boldsymbol{r}}(t)\right) = \hat{\boldsymbol{u}}(t) = \mathbf{W}_{out}\boldsymbol{q}(\hat{\boldsymbol{r}}(t)), \tag{2.12}$$

where $\mathbf{W}_{out} \in \mathbb{R}^{D\times 2N}$ is the readout matrix and $\boldsymbol{q}(\boldsymbol{r}(t)) \in \mathbb{R}^{2N}$ is designed to break the symmetry in Eq. 2.10 using the 'squaring technique' described by,

$$\boldsymbol{q}(\boldsymbol{r}(t)) = \begin{pmatrix} \boldsymbol{r}(t) \\ \boldsymbol{r}^2(t) \end{pmatrix}, \tag{2.13}$$

[2]The comments made here are adapted from [32] where Joschka Herteux conducted the literature review on the history of readout matrix designs.

where $\boldsymbol{r}^2(t) = \left(r_1^2(t), r_2^2(t), \ldots, r_N^2(t)\right)^T$. From Eq. (2.13) it then follows that Eq. (2.12) can be rewritten as,

$$\begin{aligned}\hat{\boldsymbol{\psi}}\left(\hat{\boldsymbol{r}}(t)\right) = \mathbf{W}_{out}\boldsymbol{q}(\hat{\boldsymbol{r}}(t)) &= \left(\mathbf{W}_{out}^{(1)}\,\mathbf{W}_{out}^{(2)}\right)\begin{pmatrix}\hat{\boldsymbol{r}}(t)\\ \hat{\boldsymbol{r}}^2(t)\end{pmatrix}\\ &= \mathbf{W}_{out}^{(1)}\,\hat{\boldsymbol{r}}(t) + \mathbf{W}_{out}^{(2)}\,\hat{\boldsymbol{r}}^2(t),\end{aligned} \tag{2.14}$$

where $\mathbf{W}_{out}^{(1)}$ is the 'linear readout matrix', and $\mathbf{W}_{out}^{(2)}$ is the 'square readout matrix' which breaks the symmetry in Eq. (2.10).

Ridge regression

In this Thesis $\mathbf{W}_{out}$ is determined by the ridge regression technique which consists of minimising the following expression with respect to $\mathbf{W}_{out}$,

$$\frac{1}{t_t - t_l}\left[\sum_{i=t_l}^{t_t} ||\mathbf{W}_{out}\,\boldsymbol{q}(\boldsymbol{r}[i]) - \boldsymbol{u}[i]||_2^2 + \beta\,||\mathbf{W}_{out}||_2^2\right], \tag{2.15}$$

where $\boldsymbol{r}[i]$ and $\boldsymbol{u}[i]$ are discrete-time samples of the continuous-time variables $\boldsymbol{r}(t)$ and $\boldsymbol{u}(t)$ at discrete time $i = t/\tau$ where in this Thesis τ is chosen as the time step of the integration. The corresponding time series, $[\boldsymbol{r}[i]]_{i=t_l}^{t_t}$ and $[\boldsymbol{u}[t]]_{i=t_l}^{t_t}$, are constructed by sampling $\boldsymbol{r}(t)$ and $\boldsymbol{u}(t)$ at intervals of length τ (resulting in discrete-time intervals of length 1). To be more specific, the time series are constructed as $[\boldsymbol{r}(t)]_{i=t_l}^{t_t} = [\boldsymbol{r}[t_l], \boldsymbol{r}[t_l+1], \boldsymbol{r}[t_l+2], \ldots, \boldsymbol{r}[t_t]]$ and similarly for $\boldsymbol{u}(t)$ where the discrete-times $t_l = t_{\text{listen}}/\tau$ and $t_t = t_{\text{train}}/\tau$. β is the regularisation parameter and the purpose of the $\beta\,||\mathbf{W}_{out}||_2$ term in Eq. (2.15) is to modify the linear least-squares regression to reduce the magnitudes of elements in $\mathbf{W}_{out}$ in order to discourage overfitting.

Minimising Eq. (2.15) involves taking the partial derivative of Eq. (2.15) with respect to $\mathbf{W}_{out}$ and setting the resulting expression equal to 0. This results in the following equation where the $\mathbf{W}_{out}$ that minimises Eq. (2.15) is given by,

$$\mathbf{W}_{out} = \mathbf{Y}\mathbf{X}^T\left(\mathbf{X}\mathbf{X}^T + \beta\,\mathbf{I}\right)^{-1}, \tag{2.16}$$

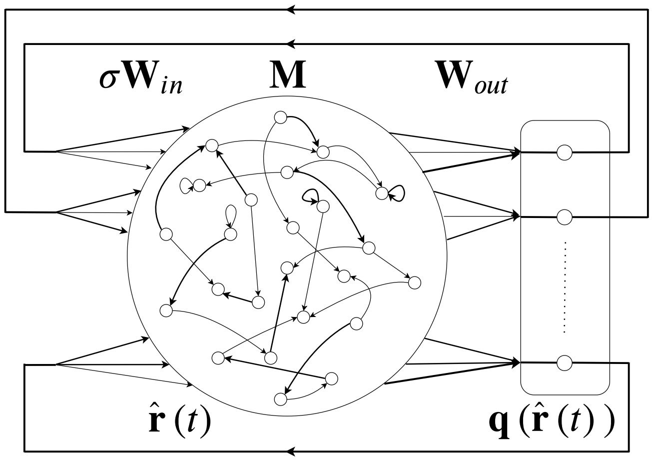


Figure 2.2: Predicting RC: The trained readout layer $\mathbf{W}_{out}\,\boldsymbol{q}(\hat{\boldsymbol{r}}(t))$ replaces the external driving input from the listening and training stages.

where,

$$\mathbf{X} = \left[\begin{array}{cccc} \boldsymbol{q}(\boldsymbol{r}[t_l]) & \boldsymbol{q}(\boldsymbol{r}[t_l+1]) & \cdots & \boldsymbol{q}(\boldsymbol{r}[t_t]) \end{array}\right], \tag{2.17}$$

is the 'response data matrix' and,

$$\mathbf{Y} = \left[\begin{array}{cccc} \boldsymbol{u}[t_l] & \boldsymbol{u}[t_l+1] & \cdots & \boldsymbol{u}[t_t] \end{array}\right], \tag{2.18}$$

is the 'input data matrix' and $\mathbf{I}$ is the identity matrix of the appropriate size.

2.3.3 Predicting stage

In the predicting stage solutions of the 'predicting RC' are computed according to the following autonomous dynamical system,

$$\dot{\hat{\boldsymbol{r}}}(t) = \gamma\left[-\hat{\boldsymbol{r}}(t) + \tanh\left(\,\mathbf{M}\,\hat{\boldsymbol{r}}(t) + \sigma\mathbf{W}_{in}\mathbf{W}_{out}\,\boldsymbol{q}(\hat{\boldsymbol{r}}(t))\,\right)\right]. \tag{2.19}$$

If the training is successful then, according to Definition 2.2.3, when Eq. (2.19) is initialised with $\hat{\boldsymbol{r}}(0) = \boldsymbol{r}(t_{\text{train}})$ the long-term dynamics of the reconstructed attractor, $\hat{\mathcal{A}}$, will resemble the long-term dynamics of $\mathcal{A}$ described by $\boldsymbol{u}(t)$ for $t > t_{\text{train}}$. An illustration of this closed-loop predicting RC setup is provided in Fig. 2.2.

2.3.4 Further comments

RC Topology

In generating many of the results presented in this Thesis, $\mathbf{M}$ is constructed with an Erdös-Renyi topology where each of the non-zero elements are then replaced with a random number between -1 and 1, the matrix is subsequently scaled to a specific spectral radius, ρ. To elaborate, $\mathbf{M}$ is designed such that each element is chosen independently to be nonzero with probability P (i.e. sparsity $= P$ or average degree $= N/P$) and these nonzero elements are chosen uniformly from $(-1, 1)$. This random sparse matrix is rescaled such that the spectral radius, which is the maximum of the absolute values of its eigenvalues, is ρ. This topology is chosen in order to provide the RC with enough dynamical flexibility to solicit multistable dynamics in tasks requiring multifunctionality.

Similarly, the input matrix, $\mathbf{W}_{in}$, is designed such that each row has only one nonzero randomly assigned element, chosen uniformly from $(-1, 1)$.

Role of $\boldsymbol{\rho}$

A key parameter involved in this RC setup is the spectral radius, ρ, of the internal connections described by $\mathbf{M}$. One of the central discoveries reported on in Chapter 6 is that multifunctionality is found to occur within a 'Goldilocks region' whereby, depending on the level of difficulty involved in a particular task, multifunctionality is achieved within a certain range of ρ values and does not occur if ρ is too large or too small. Similar results were also presented by Jiang and Lai in [78] where a given RC was found to achieve optimal performance within a specific interval of ρ values on several time-series prediction problems. The connection between these results and the results presented in this Thesis is discussed to a greater extent in Sec. 6.4.1.

RC design and parameters

It is important to highlight that there is currently no universally accepted method for choosing, *a priori*, any given combination of RC design and training parameters (N, P, ρ, σ, γ, β, t_{listen}, and t_{train}) in order for the RC to solve a given task.

To combat this, algorithms have been developed which optimise these parameters in both local and global manners to reduce the error in time-series prediction problems [79, 80]. However, the drawback in using these optimisation algorithms is that the set of training parameters that are found to provide the longest short-term predictions of a given chaotic attractor may not be suitable for reconstructing the long-term dynamics of the attractor.

Instead of adapting optimisation algorithms like those mentioned above for the case of multifunctionality, throughout this Thesis there is an emphasis placed on investigating the range of certain RC design parameter values in which multifunctionality is achieved. This involves fixing most of the design parameters and inspecting the dependency between a successful training outcome and the remaining 'free' training parameters. This method provides a map/grid whereby the relationship between certain parameters and multifunctionality are identified in relation to a specified error criterion.

2.3.5 Other RC Formulations

In this Thesis, the performance of the continuous-time RC in Eq. (2.19) on tasks involving multifunctionality is assessed and compared with other RC formulations to determine the advantages or disadvantages posed by each of these RC designs. These include the original ESN formulation introduced in [29] and the recently developed next-generation RC setup presented in [81]. The steps involved in training these RCs to perform attractor reconstruction are outlined below, the same ridge regression approach outlined in Sec. 2.3.2 is used to train both of these RCs. Furthermore, additional examples of RC formulations which are not explored in this Thesis are also highlighted.

Original ESN

The continuous-time listening RC in Eq. (2.10) is derived from the original discrete-time formulation described by Jaeger in [29] and is given by rewriting

$\dot{\boldsymbol{r}}(t)$ as,

$$\dot{\boldsymbol{r}}(t) = \frac{\boldsymbol{r}[i+1] - \boldsymbol{r}[i]}{\tau}, \tag{2.20}$$

where τ is the same integration time step used in computing solutions of the continuous-time RC and $\boldsymbol{r}[i]$ is the state of the discrete-time RC at discrete time i. From Eq. (2.20) it follows that the state of this discrete-time RC evolves according to,

$$\boldsymbol{r}_{LI}[i+1] = (1-\eta)\boldsymbol{r}_{LI}[i] + \eta \tanh\left(\mathbf{M}\,\boldsymbol{r}_{LI}[i] + \sigma \mathbf{W}_{in}\,\boldsymbol{u}[i]\right), \tag{2.21}$$

where $\eta = \gamma\tau$ and $i = t/\tau$. The parameter $\eta \in [0, 1]$ is known as the 'leaky-integrator' parameter and is used to control the amount of new information that the RC receives on each iteration of Eq. (2.21). As a result, this discrete-time RC is known as the leaky-integrator RC (LI-RC), hence the relabelling of $\boldsymbol{r}[i]$ as $\boldsymbol{r}_{LI}[i]$. In the original ESN design there is no contribution from the leaky integrator term, which is the same as writing Eq. (2.21) with $\eta = 1$, resulting the following update equation,

$$\boldsymbol{r}[i+1] = \tanh\left(\mathbf{M}\,\boldsymbol{r}[i] + \sigma \mathbf{W}_{in}\,\boldsymbol{u}[i]\right). \tag{2.22}$$

Next Generation RC

Recently, Gauthier *et al.* introduced the 'Next Generation RC' (NG-RC) in [81] which differs significantly from the RC formulations described so far. The NG-RC is constructed as a linear reservoir with nonlinear readout and in [81] it is shown to achieve remarkable results in time series prediction problems.

When using this setup for time-series prediction problems, the input data are transformed with a 'polynomial multiplication dictionary', $\mathbf{P}^{[O]}$, into a higher dimensional state space consisting of the unique polynomials of orders, O. For example, transforming a two-dimensional input data point, $\boldsymbol{u}[i] = (u_1[i], u_2[i])^T$, with $\mathbf{P}$ for $O = 1, 2$ is written as,

$$\mathbf{P}^{[1,2]}(\boldsymbol{u}[i]) = \left(u_1[i]\ u_2[i]\ u_1^2[i]\ u_2^2[i]\ u_1[i]u_2[i]\right)^T. \tag{2.23}$$

A 'time shift expansion dictionary', $\mathbf{L}_k^s$, is used in [81] to shift the input data in time in order to distinguish the NG-RC from nonlinear vector autoregression algorithms [82]. The k value defines the number of past data points that the current data point is concatenated with, and the s value denotes how far these points are separated in time. Following the example of a two-dimensional input data point, this time shift is written as,

$$\mathbf{L}_2^1(\boldsymbol{u}[i]) = (u_1[i]\ u_2[i]\ u_1[i-1]\ u_2[i-1])^T . \tag{2.24}$$

The NG-RC's response to $\boldsymbol{u}[i]$ during training is written as,

$$\boldsymbol{r}_{NG}[i+1] = \mathbf{P}^{[O]}(\mathbf{L}_k^s(\boldsymbol{u}[i])). \tag{2.25}$$

Therefore, in this example where $O = 1, 2$, $k = 2$, and $s = 1$, it follows that the corresponding RC state consists of $N = 14$ 'neurons' where $\boldsymbol{r}_{NG}[i+1] \in \mathbb{R}^{14}$ is written as,

$$\Big(u_1[i]\ \ u_2[i]\ \ u_1[i-1]\ \ u_2[i-1]\ \ u_1^2[i]\ \ u_1[i]u_2[i]\ \ u_1[i]u_1[i-1]\ \ u_1[i]u_2[i-1]$$
$$u_2^2[i]\ \ u_2[i]u_1[i-1]\ \ u_2[i]u_2[i-1]\ \ u_1^2[i-1]\ \ u_1[i-1]u_2[i-1]\ \ u_2^2[i-1]\Big)^T . \tag{2.26}$$

In this setup, the reservoir state vector, $\boldsymbol{r}_{NG}[i]$, is projected using a readout matrix, $\mathbf{W}_{out}$, to resemble, $\Delta\boldsymbol{u}[i] = \boldsymbol{u}[i] - \boldsymbol{u}[i-1]$, for $i > i_{train}$ where i_{train} is the end of the training time. $\mathbf{W}_{out}$ is found using Eq. (2.16). The input training data $\mathbf{Y} = [\boldsymbol{u}[i_{warm}], \ldots, \boldsymbol{u}[i_{train}]]$ is transformed to the state matrix $\mathbf{X} = \boldsymbol{q}\left(\mathbf{P}^{[O]}(\mathbf{L}_k^s(\mathbf{Y}))\right)$ where the same $\boldsymbol{q}(\cdot)$ from Eq. (2.13) is used here but it is not necessary to do so in all cases. Note, a warm up time of $i_{warm} = ks$ is needed, where entries of the state matrix at time $i < i_{warm}$ are not defined. The output target matrix used in Eq. (2.16) for this setup is written as, $\mathbf{Y}' = \mathbf{Y}[i] - \mathbf{Y}[i-1]$ in accordance with how $\Delta\boldsymbol{u}[i]$ is written. The trained NG-RC evolves according to,

$$\hat{\boldsymbol{r}}_{NG}[i+1] = \mathbf{P}^{[O]}(\mathbf{L}_k^s(\hat{\boldsymbol{u}}_{NG}[i-1] + \mathbf{W}_{out}\boldsymbol{q}\left(\hat{\boldsymbol{r}}_{NG}[i]\right))), \tag{2.27}$$

for $\hat{\boldsymbol{r}}_{NG}[0] = \boldsymbol{r}_{NG}[i_{train}]$ and $\hat{\boldsymbol{u}}_{NG}[i-1] = \boldsymbol{u}[i_{train}] + \sum_{i_{train}}^{i-1} \mathbf{W}_{out}\boldsymbol{q}\left(\hat{r}_{NG}[i]\right)$ estimates $\boldsymbol{u}[i]$ for $i > i_{train}$.

As the NG-RC requires tuning only O, k, s, and β, the issues which relate to improper random initialisations of $\mathbf{M}$ and $\mathbf{W}_{in}$ do not arise. On the other hand, the results shown in Chapter 7 shed some light on the limitations to what can be achieved with the NG-RC because of its design in comparison to CT and LI-RCs on a benchmark problem for multifunctionality.

Additional examples of RC designs

Since the introduction of ESNs and LSMs, there have been many variations of RCs both in terms of mathematical formulation and reservoir design. In this section, many of these developments are highlighted, however, this is not an exhaustive list of all RC designs.

Appeltant *et al.* presented a single-node time-delay based RC in [83]. Gallicchio *et al.* assembled a 'deep RC' consisting of many reservoir layers in [84]. New formulations have combined these time-delay and deep RC philosophies to produce deep time-delay RCs [85]. Carroll and Pecora studied how changes in reservoir topology effects the performance of polynomial and hyperbolic tangent based RC architectures in [86].

Based on the promise of ML and data-driven models to 'fill the gaps' of inaccuracies present in knowledge-based models of physical processes, in [87] Pathak *et al.* combine the best of both worlds in order to produce a 'hybrid RC' which is a RC that is designed to incorporate knowledge on the physical process it is trained to model. This new reservoir computing paradigm is shown to more accurately predict how a given process' state variables evolve for a much longer period of time than either of its constituents alone. Since this hybrid RC approach was introduced, a variation called the 'Combined Hybrid-Parallel Prediction' (CHyPP) method was presented in [88] which has recently been combined with the Atmospheric Global Circulation Model (AGCM) in [89] to produce improved longer term forecasts on how certain atmospheric state variables evolve in time.

Physical RCs

The premise of the RC approach to ML is for the internal dynamics of the RC to provide a response to a history of driving inputs without needing to alter its structure throughout the training. Based on this many physical mediums have been used to play the role of an internal layer to a RC by taking advantage of some material properties. Various constructions of these 'Physical Reservoir Computers' have been developed which include optoelectronic systems [90], neuromorphic chips [91], nanomagnetic systems [92], and examples of *in materio* designs such as protein molecules [93] and concrete building blocks [94], octopus inspired soft robotic arm [95]. For recent reviews and discussions which provide further insight on the scope of physical RCs, see [96–100].

2.4 Attractor Reconstruction

The RC training procedure outlined in the Sec. 2.3 to train the continuous-time RC defined in Eq. (2.10) is now applied to the task of reconstructing the dynamics of the chaotic Lorenz attractor [75]. Trajectories on this attractor, $\mathcal{L}$, are generated by computing solutions of the following set of equations,

$$\begin{aligned} \dot{x}_1(t) &= 10\left(x_2(t) - x_1(t)\right), \\ \dot{x}_2(t) &= x_1(t)\left(28 - x_3(t)\right) - x_2(t), \\ \dot{x}_3(t) &= x_1(t)\, x_2(t) - \frac{8}{3} x_3(t), \end{aligned} \tag{2.28}$$

from the initial condition (IC), $(x_1(0), x_2(0), x_3(0)) = (1, 1, 1)$, using the 4^{th} order Runge-Kutta method with time step $\tau = 0.01$.

The three dynamical variables describing trajectories on $\mathcal{L}$ are used to provide the driving input to the RC during the training, in this case $\boldsymbol{u}(t)$ is written as $\boldsymbol{u}_{\mathcal{L}}(t) = (x_1(t), x_2(t), x_3(t))$ and the RC's response to this input is found by generating solutions of Eq. (2.10) with the same τ used to compute trajectories on $\mathcal{L}$. The RC is trained according to the steps outlined in Sec. 2.3.2 with $\boldsymbol{q}(\boldsymbol{r}(t))$ as defined in Eq. (2.13). The RC design and training parameters for the experiments conducted in the present section are specified

N	P	σ	γ	β	t_{listen}	t_{train}
100	0.05	0.2	10	10^{-4}	100	100

Table 2.1: RC design and training parameters used to reconstruct dynamics of Lorenz attractor.

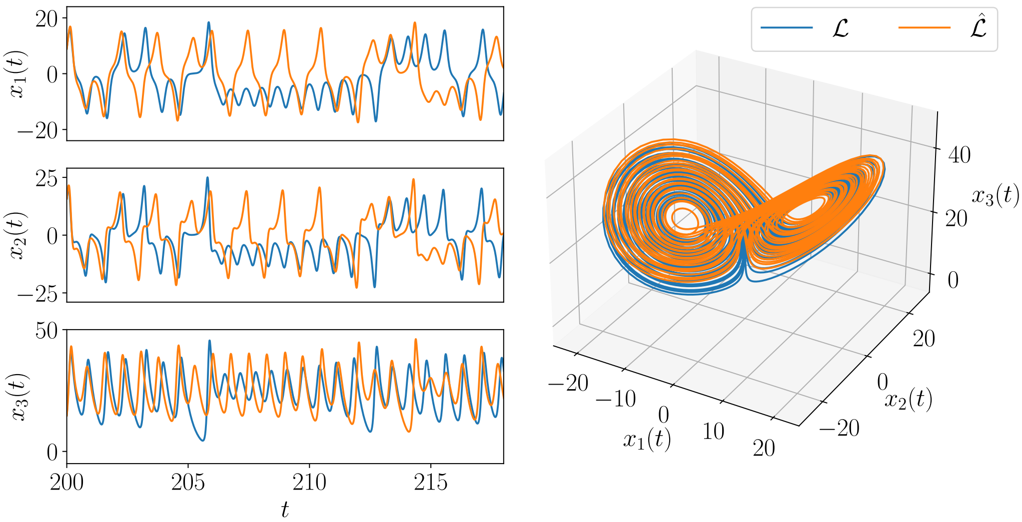


Figure 2.3: Illustration of the RC's reconstructed Lorenz attractor, $\hat{\mathcal{L}}$, compared with the true attractor, $\mathcal{L}$, for $\rho = 0.6$.

in Table 2.1 and the influence of the spectral radius, ρ, on the RC's ability to reconstruct $\mathcal{L}$ is investigated for several values of ρ in the following sections.

The result of training the RC with $\rho = 0.6$ is shown in Fig. 2.3. Here the RC's reconstruction of $\mathcal{L}$ is denoted as $\hat{\mathcal{L}}$ to be consistent with the prediction of $\boldsymbol{u}(t)$ as it is defined as $\hat{\boldsymbol{u}}(t)$ and t in this case is the time from which the listening stage begins. Trajectories on $\hat{\mathcal{L}}$ are computed by generating solutions of Eq. (2.19) initialised with $\hat{\boldsymbol{r}}(0) = \boldsymbol{r}(t_{\text{train}})$ and projecting $\hat{\boldsymbol{r}}(t)$ to $\mathbb{P}$ via Eq. (2.12). On the left of the figure are the RC's short-term prediction of all dynamical variables. Although the prediction breaks down for $t > 202.5$, which is 2.5 time units after solutions of the predicting RC are computed from its initial condition, on the right of the figure the long-term behaviour of $\hat{\mathcal{L}}$ (when $t \gg t_{\text{train}}$ in this case $t = (400, 500)$) is shown in the (x_1, x_2, x_3)-plane

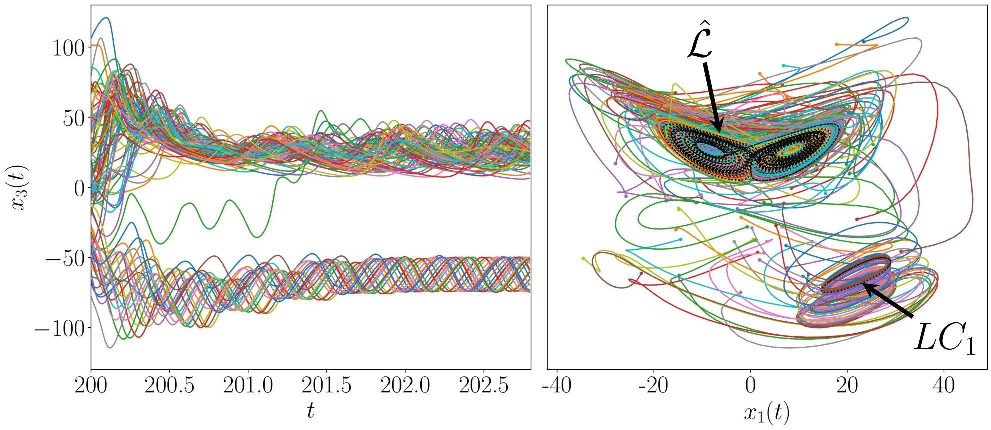


Figure 2.4: Dynamics of the predicting RC (trained to reconstruct $\mathcal{L}$) when initialised from 100 random initial conditions. Left: Time trace of the predicted x_3 variable. Right: Transient dynamics of the RC in the predicted (x_1, x_3)-plane. Dotted black lines describe a trajectory on each of the resulting attractors in $\mathbb{P}$.

and closely resembles the dynamics of the original $\mathcal{L}$.

2.5 Untrained Attractors

The results presented in this Section help to further crack open the black box by providing new insight into how a RC learns to reconstruct an attractor.

2.5.1 Global Reconstruction

While Fig. 2.3 illustrates that the RC successfully reconstructs the dynamics of $\mathcal{L}$, the next question to consider is: how well did the RC reconstruct the global dynamics of the Lorenz system described in Eq. (2.28)? As $\mathcal{L}$ is the only stable attractor present in the state space of Eq. (2.28) then the same should be the case for $\hat{\mathcal{L}}$ in $\mathbb{P}$. However, it turns out that this is not always the case.

2.5.2 Probing $\mathbb{P}$

In order to shed some light on the above and gain insight towards the structure of $\mathbb{P}$, solutions of the same predicting RC, as discussed in the previous section, are generated when this RC is initialised from 100 random ICs. The resulting dynamics are shown in Fig. 2.4 which, to much surprise, reveals that $\hat{\mathcal{L}}$ is not alone in $\mathbb{P}$, there is also a limit cycle (LC_1) present which was not part of the training and will be referred to as an 'untrained attractor' (UA).

In the left hand side of Fig. 2.4 it is shown that within ≈ 1.5 time units of the simulation, the state of the predicting RC has settled down to either attractor and on closer inspection it is found that the state of the RC tends towards LC_1 from 23 of these 100 random ICs.

This interesting result is further exemplified in the right hand side of Fig. 2.4 where the transient dynamics in the (x_1, x_3)-plane provides an indication of how $\mathbb{P}$ is shared between $\hat{\mathcal{L}}$ and LC_1. From this perspective the basin boundary between $\hat{\mathcal{L}}$ and LC_1 in the (x_1, x_3)-plane is not a straight line dividing $\mathbb{P}$. In certain regions of the (x_1, x_3)-plane there appears to be isolated nests of ICs which are arbitrarily close together from which the RC tends to different attractors. Furthermore, there are a number of ICs that are relatively close to LC_1 from which the RC subsequently tends to $\hat{\mathcal{L}}$. While it is only in $\mathbb{S}$ that the structure of the actual basins can be determined, interpreting the basin structure in $\mathbb{P}$ provides further insight into the dynamical properties of the predicting RC.

UAs for $\rho = [0.2, 0.65]$

The role of UAs and their coexistence with the desired reconstructed attractor in $\mathbb{P}$ is not widely reported in the literature. In this Thesis, bifurcation analyses of UAs in $\mathbb{P}$ provide vital insight in assessing how the RC solves a given task and becomes multifunctional. Without introducing significant distraction from the main focus of this Thesis, which is to explore the dynamics and applications of multifunctional RCs, the above experiment is repeated with the same set of random ICs for $\rho = [0.2, 0.65]$ in order to showcase the wider influence of UAs on the RC's ability to reconstruct $\mathcal{L}$. The resulting dynamics of the RC for these ρ values are illustrated in Fig. 2.5 where the same randomly initialised $\mathbf{M}$ and $\mathbf{W}_{in}$, that were used to generate the results shown so far, are used in each of these simulations and $\mathbf{M}$ is rescaled to have a particular spectral radius, ρ.

It is clear from Fig. 2.5 that there is a strong relationship between ρ, UAs, and the reconstruction of $\mathcal{L}$. By increasing ρ from the previous example of $\rho = 0.6$ to 0.65, LC_1, becomes unstable and $\hat{\mathcal{L}}$ is the only stable attractor present in $\mathbb{P}$. By decreasing ρ from 0.6, the influence of LC_1 also diminishes as the number of random ICs from which the state of the RC tends towards LC_1 decreases from 20 when $\rho = 0.55$ to 6 when $\rho = 0.35$ until LC_1 becomes unstable for $\rho < 0.35$. Even though this results in an increase in the number of random ICs from which the RC's state settles to $\hat{\mathcal{L}}$, there is a significant difference in the lengths of the transient time from some of these ICs. This is due to the presence of another UA, a limit cycle, LC_2, which becomes stable as ρ is reduced from 0.4 to 0.35. There is a coexistence between these three attractors, $\hat{\mathcal{L}}$, LC_1, and LC_2 for $\rho = 0.35$ with the state of the RC converging to LC_2 from 83 of the random ICs and $\hat{\mathcal{L}}$ from the remaining 11 random ICs. This is in stark contrast to the dynamics of the RC when $\rho = 0.4$ as 90 of the random ICs ended up on $\hat{\mathcal{L}}$. As ρ is further decreased, LC_2's basin of attraction continues to increase in volume as 94 of the random ICs are in this basin for $\rho = 0.35$.

Untrained Chaotic Attractor

From additional investigation, it was found that when decreasing ρ from 0.35 like in Fig. 2.5, before LC_1 disappears from $\mathbb{P}$ as shown for $\rho = 0.3$, LC_1 becomes a chaotic attractor, CA_1, through an infinite sequence of period doubling bifurcations. In Fig. 2.6, the coexistence between CA_1, LC_2, and $\hat{\mathcal{L}}$ is shown for $\rho = 0.3418$ when the RC was initialised with the same set of random ICs used in the experiment detailed in Fig. 2.5. The state of the RC approaches CA_1 from only 4 of these random ICs and CA_1 subsequently becomes unstable for ρ slightly less than 0.3418. Despite the small change in ρ from 0.35 to 0.3418, LC_2's basin of attraction goes on to grow further in

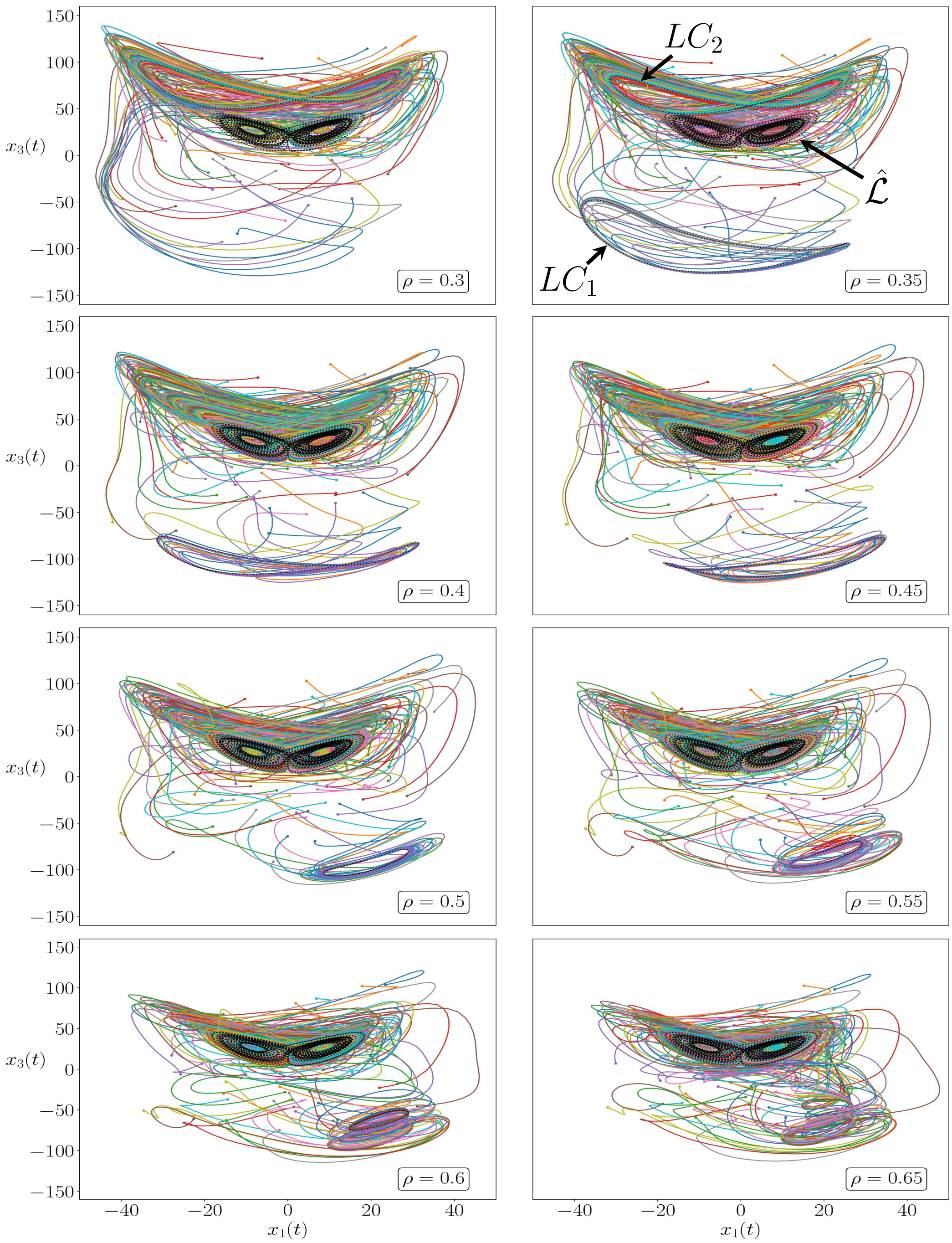


Figure 2.5: Dynamics of the predicting RC (trained to reconstruct $\mathcal{L}$) in the (x_1, x_3)-plane of $\mathbb{P}$ when initialised from 100 random initial conditions with ρ values as specified in each example. Dotted black lines describe a trajectory on each of the resulting attractors.

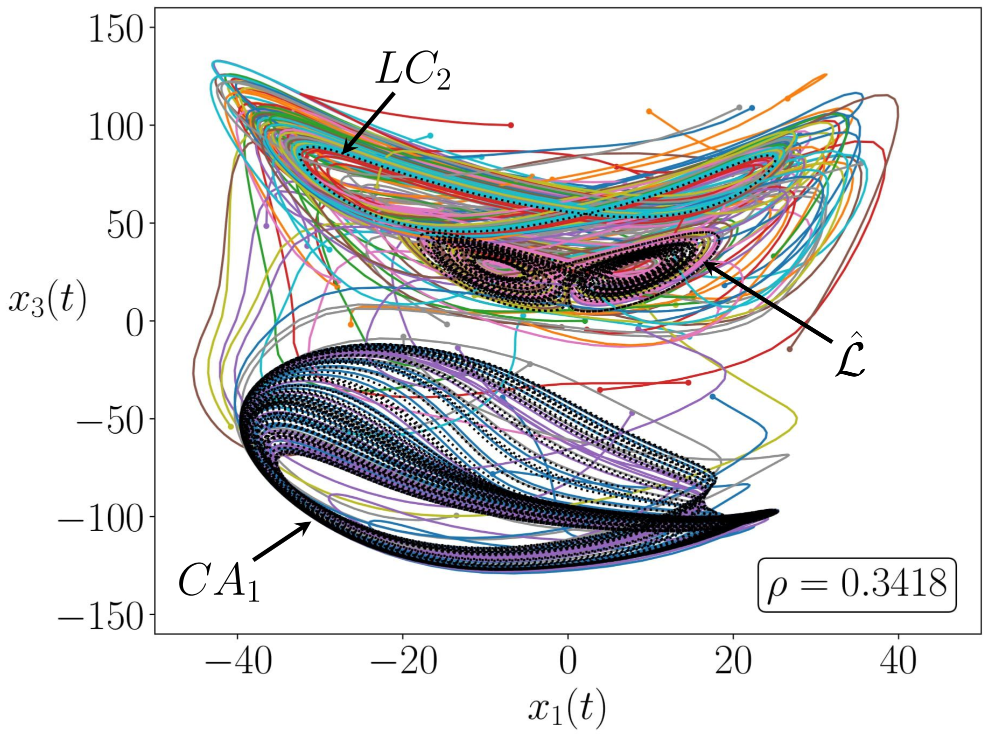


Figure 2.6: Dynamics of the predicting RC (trained to reconstruct $\mathcal{L}$) in the (x_1, x_3)-plane of $\mathbb{P}$ when initialised from 100 random initial conditions with $\rho = 0.3418$. Dotted black lines describe a trajectory on each of the resulting attractors, CA_1, LC_2, and $\hat{\mathcal{L}}$.

volume with 86 of the random ICs now in its basin.

2.5.3 General comments on UAs

While the bifurcations responsible for the changes in the stability of these UAs have not been determined, it is important to note from Fig. 2.5 that after a given UA becomes unstable, a signature of its previously stable dynamics is present in the RC's transient dynamics for ρ values long after the associated bifurcation takes place. In addition, prior to a given UA becoming unstable it is found that less and less of the random ICs are within its basin of attraction. While the RC is able to reconstruct the dynamics of $\mathcal{L}$ for a broad range of ρ values, it is clear that $\hat{\mathcal{L}}$ is not always globally stable. As the RC's trajectories remain bounded due to its design in Eq. (2.19), the UAs can also be thought of as playing an important role by providing the state of the RC with a route to exhibiting some stable dynamics before $\hat{\mathcal{L}}$ becomes globally stable.

In the context of the black-box problem in ML discussed in Sec. 2.1.2 and Appendix A, these UAs can be seen as unintended functionalities of the trained RCs. UAs are likely to be a widespread phenomenon which play an integral role in how a RC learns. At the same time UAs pose an inherent problem in terms of training a RC to become a digital twin of a dynamical system like in [101] and obstruct a RC's capacity to reconstruct global dynamics from limited local information of a given dynamical system like in [102]. Despite these negative implications, there are further lessons to be learned from the existence of UAs and these are discussed in the following section.

2.6 Reconstruction Capacity

Within the context of attractor reconstruction, the following question naturally emerges, what is the reconstruction capacity of a given RC? In other words, how many attractors can a given RC reconstruct?

While UAs are indeed an undesirable outcome of the training procedure, their existence also provides a new perspective towards answering the question posed above. If the nature of these UAs were instead seen to serve as additional operational states of the RC, then the results shown in Figs. 2.4-2.6 can also be reinterpreted as a multistable RC with multiple modes of computation. In this instance, changing the state of the RC is sufficient in order to switch between each of these computational modes without the need of altering the RC's connections. This gives rise to the following question, can a RC be trained to specifically perform more than one task without the need of retraining $\mathbf{W}_{out}$ for each individual task? In other words, can a RC be trained have more than one function and become multifunctional?

The above question is the central topic of this Thesis and in order to shed some light on this, inspiration is drawn from the behaviour of 'multifunctional neural networks' present in the brain of humans and other organisms.

Chapter 3

Multifunctional Reservoir Computing

In the pursuit of developing artificially intelligent systems, there is much to be gained from dually integrating further physiological features of BNNs and knowledge of dynamical systems into machine learning environments. In this Chapter, such a two-armed approach is employed in order to translate the main characteristic features of multifunctionality from biological to artificial neural networks via reservoir computing.

3.1 Multifunctionality in the Brain

Definition 3.1.1 (Multifunctionality)**.** *Multifunctionality is the term used to describe neural networks that are capable of changing their dynamics on demand of a given duty without needing to alter their synaptic properties. See Fig. 3.1 for an illustrative description of this phenomenon.*

Multifunctionality is an essential element of certain BNNs and has been an active area of research in neuroscience since the mid-1980s with seminal work published by Mpitsos and Cohan in [23] and Getting in [24] followed by further review papers [25, 26] and more recently reviewed by Briggman and Kristan Jr. in [22].

The emergence of multifunctionality as a recognised feature of neuronal architectures improves upon previous notions that considered the brain to consist of many neural networks which, on their own, are dedicated to performing individual tasks. The existence of multifunctionality also indicates that separate neural networks are not needed by the brain in order for it to execute a certain behaviour or for each modification thereof.

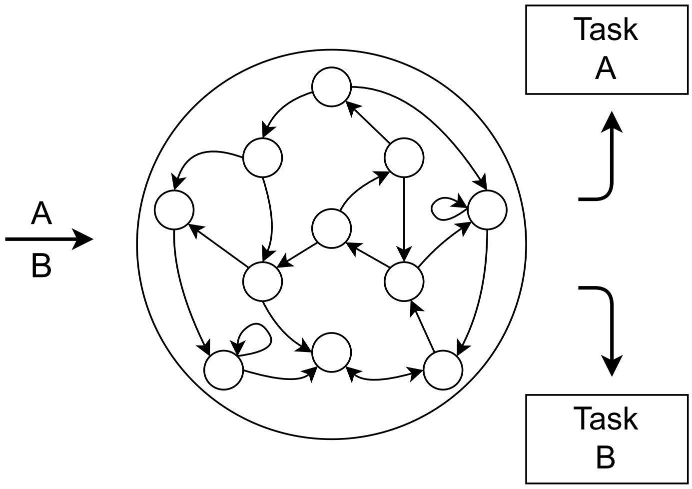


Figure 3.1: Illustration of the basic working features of a multifunctional neural network. For each input (A or B) the network executes a specific task (Task A or Task B) without changing any network connections.

3.1.1 Examples in nature

The schematic in Fig. 3.1 illustrates the basic working features of a multifunctional neural network. This diagram highlights that, based on a particular task related input, A or B, a multifunctional neural network will perform either Task A or Task B. Examples of the different tasks that multifunctional neural networks are known to perform are outlined below.

Multifunctionality in animals

In the seminal examples of multifunctional neural networks mentioned previously, it was shown in [23] that by modulating an input to a network located in the brain of the carnivorous sea slug, *Pleurobranchaea californica*, this gave rise to a switching between ingestion and egestion oral behaviours.

Furthermore, it was found that a subset of the same bundle of neurons in the brain of the medicinal leech, *Hirudo medicinalis*, can switch their activity pattern, once a change in its surroundings is sensed, to drive either a swimming or crawling motion [103]. Similar switching between swimming and crawling has also been found in a multifunctional neural network located in the brain of the marine mollusk, *Tritonia diomedea*, in [104].

Multifunctionality in humans

Evidence of multifunctionality has also been observed in the human brain. In [27] it is reported that a cluster of neurons located in the *pre-Bötzinger complex*, a region of the mammalian brain stem, is responsible for regulating a switching between different respiratory patterns. Depending on a particular input, these neurons can alter their activity pattern accordingly to elicit a switching between eupneic (regular) breathing, sighing or gasping.

Origins of Multifunctionality

Multifunctionality is a fundamental feature of many neuronal architectures and considered to be key to the survival of certain species over time. All in all, multifunctionality is typically found in networks with a small number of neurons. Based on this it is argued that multifunctionality may naturally emerge from an efficient use of limited resources, which in this case is a short supply of neurons [22]. What is more is that the ability of a given neural network to become multifunctional can be considered as an evolutionary advantage in enduring environmental changes, thereby reflecting the developmental history of certain organisms.

3.1.2 Dynamical principles

In all uses of the term multifunctionality, the common factor is that multifunctional neural networks are described as distinct pools of neurons capable of performing a multitude of mutually exclusive tasks.

Multistability

What is ubiquitous amongst the examples of multifunctionality in nature described in Sec. 3.1.1 is that these networks in principle resemble a multistable system. Multistability is the term that is used to describe a dynamical system that has a coexistence of multiple stable attractors in its state space and is a common feature of numerous dynamical systems. For an overview on the history of multistability and examples of its prominence across several disciplines of science, see [105].

Based on a particular input, there is a distinct activity pattern expressed by the neurons in the network in order to perform one of the many tasks required of it. When needed, neurons in these multifunctional networks switch to another activity pattern to collectively execute a different task while the network connections remain fixed. Based on this, it is natural to consider that the behaviour of a multifunctional neural network is analogous to the dynamics of a multistable dynamical system whereby for each task that the network performs there is, in this sense, an attractor associated with it. This attractor is in coexistence with several other attractors in the networks state space, each of which are distinctly related to the tasks that the network performs.

In the context of the examples mentioned in Sec. 3.1.1, a multifunctional neural network performs a multitude of mutually exclusive tasks by providing a coexistence of: eupneic, sighing, and gasping attractors in the case of the *pre-Bötzinger complex*, swimming and crawling attractors in the case of *Tritonia diomedea* and *Hirudo medicinalis*, and ingestion and egestion attractors in *Pleurobranchaea californica*.

This multistability interpretation of events dates back to the seminal work of Mpitsos and Cohen in [23], where the researchers use the term multifunctionality to describe networks in which neurons have multiple dynamical regimes but without altering the strength of the synapses. This type of behaviour is akin to a system with a coexistence of attractors. More recently, in [22], there is a strong emphasis placed on interpreting the phenomenon of multifunctionality through the language of dynamical systems, in particular describing multifunctionality in connection with multistable neural dynamics.

In this Thesis, the focus is on translating multifunctionality to the reservoir computing ML paradigm via the above prism of multistability.

Other dynamical systems associations

The term multifunctionality has at times been used in association with different dynamical phenomena. For example, in [24] the term is used to describe networks that are 're-configurable' meaning that they can have multiple modes of operation by changing properties of certain synapses or neurons which is in contrast to multifunctionality as defined in Definition 3.1.1. In this example, by considering the network to be a dynamical system this description is analogous to an attractor undergoing a bifurcation or having its dynamics distorted by changing parameters in the system.

Further examples where the term multifunctionality was used in a context outside of multistability can be found in [25, 26] where the term is used to describe the different behaviours generated by the coupling of networks. This again is in contrast to Definition 3.1.1 as only a single network is required to perform multiple tasks without coupling to other networks.

3.1.3 From biological to artificial neural networks

The characterisation of neuronal behaviour regularly invokes the language of dynamical systems [22, 106–108]. Furthermore, conceptualising certain traits of neurons through the lens of dynamical systems can act as a bridge between the worlds of biological and artificial neural networks where advancements in one can contribute to the other.

Bridging the gap

In this Thesis, it is through connecting the principal characteristic of multifunctionality in BNNs to a mathematical metaphor of multistability that allows multifunctionality to be realised in a ML environment via reservoir computing. However, careful consideration is needed to achieve this as many conceptual complications arise if multifunctionality is to be treated as directly analogous to the phenomenon of multistability in dynamical systems. Nevertheless, it is through exploring the limits of multifunctionality in paradigmatic scenarios that these philosophical issues are resolved and as a result many practical applications of multifunctionality are identified and explored throughout this Thesis.

Key concept

As a starting point, it is by recognising that a multifunctional neural network can in principle resemble a system with a coexistence of attractors, and that a RC can be trained to reconstruct the dynamics of an attractor as shown in Sec. 2.4, the following question practically manifests itself: can a RC be trained to reconstruct a coexistence of attractors? Therefore, in the spirit of the claims made during this Chapter, such a RC can be said to exhibit multifunctionality.

In the next section, the steps involved in training a RC to become multifunctional and reconstruct a coexistence of several attractors are outlined.

3.2 Multifunctionality in a RC

Having outlined the parallels between multifunctionality and multistablity in previous sections, the mathematics that describes how multifunctionality can be achieved with a RC is developed in Sec. 3.2.1. Following this a new training technique for the continuous-time RC setup described in Eqs. (2.10) and (2.19) is developed in Sec. 3.2.2 as a means to realise multifunctionality in an artificial setting. This new training step is also scalable to other RC formulations mentioned in Sec. 2.3.5. The characteristics of multifunctionality in a RC are compared with related works in Sec. 3.3 and the advantages of multifunctionality are outlined in Sec. 3.4.

3.2.1 Achieving multifunctionality

In order for a RC to achieve multifunctionality in the context of attractor reconstruction it is necessary for it to, in some way, inherit the ability to **perform multiple tasks using the same readout matrix**, $\mathbf{W}_{out}$, **and without changing any other properties of the RC**.

The connection between a RC achieving attractor reconstruction and the concept of generalised synchronisation (GS), as discussed in Sec. 2.2.2, is now extended to the case of multifunctionality where GS is required to occur between a listening RC like in Eq. (2.4) and two or more drive systems in the form of Eq. (2.1).

Extending Generalised Synchronisation

Consider two vectors, $\boldsymbol{u}_1(t) \in \mathbb{R}^D$ and $\boldsymbol{u}_2(t) \in \mathbb{R}^D$, that describe trajectories on two different attractors, $\mathcal{A}_1 \subset \mathbb{R}^D$ and $\mathcal{A}_2 \subset \mathbb{R}^D$, that are from two different D-dimensional dynamical systems. Let's also consider a single listening RC described according to Eq. (2.4) where on different occasions $\boldsymbol{u}_1(t)$ and $\boldsymbol{u}_2(t)$ are used as input to drive a response from the state of the listening RC from $t = 0$ to $t = t_{\text{train}}$, these responses are written as $\boldsymbol{r}_1(t) \in \mathbb{R}^N$ and $\boldsymbol{r}_2(t) \in \mathbb{R}^N$.

In order to train a RC to achieve multifunctionality then, in accordance with Eq. (2.6), training consists of finding **a single readout function**, $\hat{\boldsymbol{\psi}} : \mathbb{R}^N \to \mathbb{R}^D$, that minimises,

$$\int_{t=t_{\text{listen}}}^{t_{\text{train}}} \frac{1}{t_{\text{train}} - t_{\text{listen}}} \left[||\hat{\boldsymbol{\psi}}\left(\boldsymbol{r}_1(t)\right) - \boldsymbol{u}_1(t)||_2^2 + ||\hat{\boldsymbol{\psi}}\left(\boldsymbol{r}_2(t)\right) - \boldsymbol{u}_2(t)||_2^2 \right] dt, \quad (3.1)$$

such that predictions of both $\boldsymbol{u}_1(t)$ and $\boldsymbol{u}_2(t)$, written as, $\hat{\boldsymbol{u}}_1(t)$ and $\hat{\boldsymbol{u}}_2(t)$ are given by,

$$\hat{\boldsymbol{u}}_1(t) = \hat{\boldsymbol{\psi}}\left(\hat{\boldsymbol{r}}_1(t)\right) \text{ and } \hat{\boldsymbol{u}}_2(t) = \hat{\boldsymbol{\psi}}\left(\hat{\boldsymbol{r}}_2(t)\right), \quad (3.2)$$

for $t > t_{\text{train}}$ where $\hat{\boldsymbol{r}}_1(t) \in \mathbb{S}$ and $\hat{\boldsymbol{r}}_2(t) \in \mathbb{S}$ describe the state of the resulting closed-loop predicting RC at time t when initialised with $\boldsymbol{r}_1(t_{\text{train}})$ and $\boldsymbol{r}_2(t_{\text{train}})$. However, GS is required to occur between the listening RC and each driving system in order for this $\hat{\boldsymbol{\psi}}$ to exist.

If GS occurs between the listening RC and each driving system where $\boldsymbol{u}_1(t)$ and $\boldsymbol{u}_2(t)$ originate from then, according to Definition 2.2.1, $\exists$ two corresponding transformations $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ where each $\boldsymbol{\phi} : \mathbb{R}^D \to \mathbb{R}^N$, two corresponding manifolds $\mathcal{M}_1 = \{(\boldsymbol{u}_1(t), \boldsymbol{r}_1(t)) : \boldsymbol{r}_1(t) = \boldsymbol{\phi}_1\left(\boldsymbol{u}_1(t)\right)\}$ and $\mathcal{M}_2 = \{(\boldsymbol{u}_2(t), \boldsymbol{r}_2(t)) : \boldsymbol{r}_2(t) = \boldsymbol{\phi}_2\left(\boldsymbol{u}_2(t)\right)\}$, and two subsets $B_1 = B_{\boldsymbol{u}_1} \times B_{\boldsymbol{r}_1} \subset \mathbb{R}^D \times$

$\mathbb{R}^N$ and $B_2 = B_{\boldsymbol{u}_2} \times B_{\boldsymbol{r}_2} \subset \mathbb{R}^D \times \mathbb{R}^N$, where $\mathcal{M}_1 \subset B_1$ and $\mathcal{M}_2 \subset B_2$, and B_1 and B_2 are basins of attraction, such that trajectories of the each unidirectionally coupled system approach $\mathcal{M}_1$ and $\mathcal{M}_2$ as $t \to \infty$ $\forall$ $(\boldsymbol{x}_1(0), \boldsymbol{r}_1(0)) \in B_1$ and $\forall$ $(\boldsymbol{x}_2(0), \boldsymbol{r}_2(0)) \in B_2$ respectively.

However, in the case of multifunctionality, there is an additional property required of each $\boldsymbol{\phi}$, and correspondingly each synchronisation manifold, $\mathcal{M}$, in order for multifunctionality to occur. It is necessary that $\boldsymbol{\phi}_1(\mathcal{A}_1) \cap \boldsymbol{\phi}_2(\mathcal{A}_2) = \emptyset$ and similarly $\mathcal{M}_1 \cap \mathcal{M}_2 = \emptyset$, otherwise, $\exists$ at least one common element of $\mathcal{A}_1$ and $\mathcal{A}_2$ that is mapped to the same element in $\mathbb{R}^N$ and there is no $\hat{\boldsymbol{\psi}}$ that can map any common element $\mathbb{R}^N$ to two elements in $\mathbb{R}^D$.

Extending Definition 2.2.2 to the present case, if $\boldsymbol{\phi}_1(\mathcal{A}_1) \cap \boldsymbol{\phi}_1(\mathcal{A}_2) = \emptyset$ and both sets $\boldsymbol{\phi}_1(\mathcal{A}_1)$ and $\boldsymbol{\phi}_1(\mathcal{A}_2)$ are attracting according to the transversal Lyapunov exponent criteria discussed in Sec. 2.2.2, then for the $\hat{\boldsymbol{\psi}}$ that is computed in keeping with Eq. (3.1) and the resulting predicting RC in Eq. 2.7, $\exists$ two response attractors, $\mathcal{S}_1 \subset \mathbb{S}$ and $\mathcal{S}_2 \subset \mathbb{S}$, that are associated with $\boldsymbol{\phi}_1(\mathcal{A}_1)$ and $\boldsymbol{\phi}_1(\mathcal{A}_2)$ respectively. Here $\mathbb{S}$ is the same as in Sec. 2.2.2, the state space of the predicting RC.

Taking into account all of the above, it is now possible to extend Definition 2.2.3 to define what it means for a RC to achieve multifunctionality in this context of GS.

Definition 3.2.1 (Multifunctionality with a RC)**.** *Multifunctionality occurs if each response attractor, $\mathcal{S}_1$ and $\mathcal{S}_2$, can* ***coexist*** *in $\mathbb{S}$ so that, when these attractors are projected from $\mathbb{S}$ to $\mathbb{P}$ using $\hat{\boldsymbol{\psi}}$, the long-term dynamics of both reconstructed attractors, $\hat{\mathrm{A}}_1$ and $\hat{\mathrm{A}}_2$, resemble the long-term dynamics of $\mathcal{A}_1$ and $\mathcal{A}_2$.*

The key word in Definition 3.2.1 is **coexist**. In order for multifunctionality to occur it is necessary that $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ lift $\mathcal{A}_1$ and $\mathcal{A}_2$ into regions of $\mathbb{S}$ that facilitate a coexistence between the corresponding $\mathcal{S}_1$ and $\mathcal{S}_2$. An illustration of this process is provided in Fig. 3.2.

It is necessary to highlight that the above statement is true whether or not $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$, i.e., whether or not there is an overlap between the attractors in $\mathbb{R}^D$. This crucial concept encapsulates the subtle difference between what

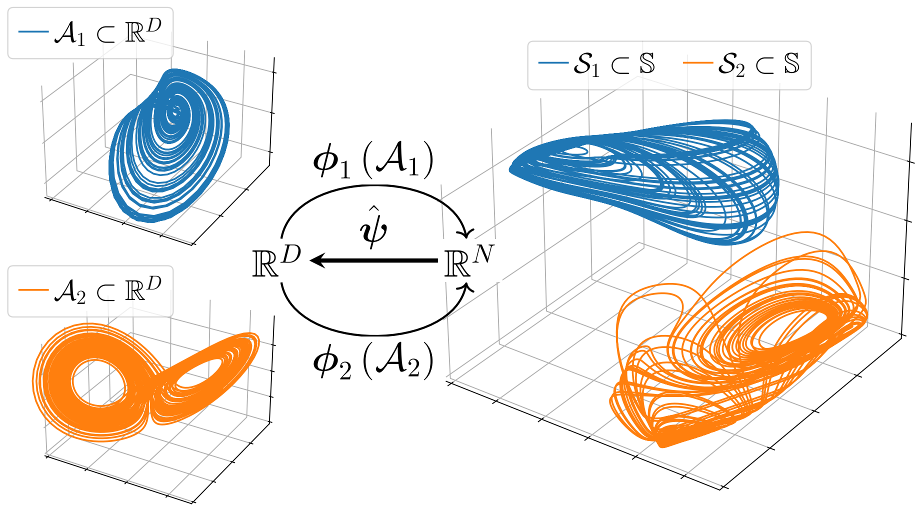


Figure 3.2: Illustration of how $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ lift the chaotic attractors, $\mathcal{A}_1 \subset \mathbb{R}^D$ and $\mathcal{A}_2 \subset \mathbb{R}^D$ (for $D = 3$), onto the coexisting chaotic attractors, $\mathcal{S}_1 \subset \mathbb{S}$ and $\mathcal{S}_2 \subset \mathbb{S}$ (plotted here using the same 3 state variables from the N-dimensional predicting RC), that can be projected back to $\mathbb{R}^D$ using $\hat{\boldsymbol{\psi}}$.

it means for a RC to exhibit multifunctionality or solely multistability. For instance, consider the case where $\mathcal{A}_1 \cap \mathcal{A}_2 \neq \emptyset$ and multifunctionality is achieved, then $\hat{\mathcal{A}}_1 \cap \hat{\mathcal{A}}_2 \neq \emptyset$. In this scenario the predicting RC is a multistable autonomous dynamical system, however, this RC produces dynamics in the D-dimensional prediction state space, $\mathbb{P}$, that no D-dimensional multistable dynamical system can.

While the sequence of statements which lead towards Definition 3.2.1 are based on attractors from two different systems, Definition 3.2.1 is still valid if, for instance, $\mathcal{A}_1$ and $\mathcal{A}_2$ are attractors from the same multistable system. Even though the driving system is the same there are two different transformations $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ created by each driving input when GS occurs with the listening RC. In this case $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$ by default which, as the results presented in Chapter 4 show, poses less problems for a RC to achieve multifunctionality as the training data describes attractors that normally coexist. The reason why Definition 3.2.1 is not constructed from this perspective is to place a greater emphasis on the critical condition that $\mathcal{S}_1$ and $\mathcal{S}_2$ are required to coexist in $\mathbb{S}$ regardless of the training data.

Furthermore, Definition 3.2.1, and the statements which precede it, can cater for training a RC to achieve multifunctionality with more than just two attractors up to any number $n \in \mathbb{N}$ of attractors, i.e., $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n$. For multifunctionality to be achieved it is necessary that the corresponding $\boldsymbol{\phi}_i\left(\mathcal{A}_i\right)$ satisfy the constraint,

$$\bigcap_{i=1}^{n} \boldsymbol{\phi}_i\left(\mathcal{A}_i\right) = \emptyset, \tag{3.3}$$

and similarly for the corresponding response attractors,

$$\bigcap_{i=1}^{n} \mathcal{S}_i = \emptyset. \tag{3.4}$$

However, there is no such constraint on each $\mathcal{A}_i$ and it is possible that,

$$\bigcap_{i=1}^{n} \mathcal{A}_i \neq \emptyset, \tag{3.5}$$

and similarly for the corresponding reconstructed attractors,

$$\bigcap_{i=1}^{n} \hat{\mathcal{A}}_i \neq \emptyset. \tag{3.6}$$

This issue relating to how multifunctionality can be achieved when $\mathcal{A}_1 \cap \mathcal{A}_2 \neq \emptyset$ is explored numerically to a great extent in Chapters 6 and 7 and it is from the insights gained through these results that multifunctionality in a RC can be exploited as a means to produce data-driven models of multistable dynamics from the epileptic brain in Chapter 8.

3.2.2 Training for multifunctionality

Generating training data

Let's consider the case of training the listening RC in Eq. (2.10) to achieve multifunctionality by reconstructing a coexistence of two attractors, $\mathcal{A}_1 \subset \mathbb{R}^D$

and $\mathcal{A}_2 \subset \mathbb{R}^D$, by driving the listening RC with input from the following vectors, $\boldsymbol{u}_1(t) \in \mathcal{A}_1$ and $\boldsymbol{u}_2(t) \in \mathcal{A}_2$ for $0 < t \leq t_{\text{train}}$. The response of the listening RC to these driving inputs is found by computing solutions of the following nonautonomous dynamical systems,

$$\dot{\boldsymbol{r}}_1(t) = \gamma \left[-\boldsymbol{r}_1(t) + \tanh \left(\mathbf{M}\, \boldsymbol{r}_1(t) + \sigma \mathbf{W}_{in}\, \boldsymbol{u}_1(t) \right) \right], \tag{3.7}$$

$$\boldsymbol{r}_1(0) = \mathbf{0}^T, \tag{3.8}$$

and,

$$\dot{\boldsymbol{r}}_2(t) = \gamma \left[-\boldsymbol{r}_2(t) + \tanh \left(\mathbf{M}\, \boldsymbol{r}_2(t) + \sigma \mathbf{W}_{in}\, \boldsymbol{u}_2(t) \right) \right], \tag{3.9}$$

$$\boldsymbol{r}_2(0) = \mathbf{0}^T, \tag{3.10}$$

using the 4^{th} order Runge-Kutta method with time step τ. The same steps involved in gathering the RC's response data during the listening and training stages described in Secs. 2.3.1 and 2.3.2 are applied. It is important to highlight that $\mathbf{M}$, $\mathbf{W}_{in}$, and all training parameters remain identical when generating the RC's response to the driving input in order to be compatible with the definition of multifunctionality from the biological perspective in Definition 3.1.1 and the reservoir computing analogue in Definition 3.2.1.

Blending Technique

The steps involved in computing $\hat{\boldsymbol{\psi}}\left(\cdot\right)$ from Sec. 2.3.2 are now adapted for the case of multifunctionality. The same readout function design as in Eq. (2.12) where $\hat{\boldsymbol{\psi}}\left(\hat{\boldsymbol{r}}(t)\right) = \mathbf{W}_{out}\, \boldsymbol{q}\left(\hat{\boldsymbol{r}}(t)\right)$ is used here, however, the ridge regression approach for computing the readout matrix, $\mathbf{W}_{out} \in \mathbb{R}^{D \times 2N}$ for the case of reconstructing only one attractor in Eq. (2.15) is modified to account for reconstructing more than one attractor. An additional 'blending' parameter, α, is also included in order to apply a different weight to the corresponding training data. This ridge regression approach consists of minimising the following expression with respect to $\mathbf{W}_{out}$,

$$\frac{1}{t_t - t_l}\left[\sum_{i=t_l}^{t_t} \alpha^2 ||\mathbf{W}_{out}\,\boldsymbol{q}(\boldsymbol{r}_1[i]) - \boldsymbol{u}_1[i]||_2^2 \right.$$
$$\left. + (1-\alpha)^2 \sum_{i=t_l}^{t_t} ||\mathbf{W}_{out}\,\boldsymbol{q}(\boldsymbol{r}_2[i]) - \boldsymbol{u}_2[i]||_2^2 + \beta\,||\mathbf{W}_{out}||_2^2 \right], \tag{3.11}$$

where $\boldsymbol{r}_1[i]$, $\boldsymbol{u}_1[i]$, $\boldsymbol{r}_2[i]$, and $\boldsymbol{u}_2[i]$ are the discrete-time samples of the continuous-time variables $\boldsymbol{r}_1(t)$, $\boldsymbol{u}_1(t)$, $\boldsymbol{r}_2(t)$, and $\boldsymbol{u}_2(t)$ at discrete-time i. The corresponding time series $[\boldsymbol{r}_1[i]]_{i=t_l}^{t_t}$, $[\boldsymbol{u}_1[i]]_{t=t_l}^{t_t}$, $[\boldsymbol{r}_2[i]]_{t=t_l}^{t_t}$, and $[\boldsymbol{u}_2[i]]_{t=t_l}^{t_t}$ are constructed using the same convention outlined in Sec. 2.3.2 by sampling $\boldsymbol{r}_1(t)$, $\boldsymbol{u}_2(t)$, $\boldsymbol{r}_2(t)$, and $\boldsymbol{u}_2(t)$ at intervals of length τ, the time step of the integration, resulting in discrete-time intervals of length 1. The regularisation parameter β plays the same role as discussed in Sec. 2.3.2 by reducing the magnitudes of elements in $\mathbf{W}_{out}$ in order to discourage overfitting.

Based on Eq. (3.11), when $\alpha = 0$ or $\alpha = 1$, one data set completely dominates the training and Eq. (3.11) reduces to Eq. (2.15). The aim is to find an α that will give rise to a $\mathbf{W}_{out}$ that enables the RC to reconstruct a coexistence of $\mathcal{A}_1$ and $\mathcal{A}_2$. The motivation behind blending the data in this way is to overcome any internal bias a given RC may have towards favouring the reconstruction of one attractor over the other during the training. This step is not necessary, however when the dynamical characteristics of $\mathcal{A}_1$ and $\mathcal{A}_2$ differ, for example if $\mathcal{A}_1$ and $\mathcal{A}_2$ describe trajectories on a limit cycle and a chaotic attractor, or for instance if $\mathcal{A}_1$ and $\mathcal{A}_2$ describe trajectories on attractors from different dynamical systems, then applying different weights to the respective attractors can help improve the outcome of the training. The relationship between multifunctionality and α is explored in Chapter 4.

As outlined in Sec. 2.3.2, minimising Eq. (3.11) with respect to $\mathbf{W}_{out}$ involves taking the partial derivative of Eq. (2.15) with respect to $\mathbf{W}_{out}$ and setting the resulting expression equal to 0. This results in the following equation where the $\mathbf{W}_{out}$ that minimises Eq. (3.11) is given by,

$$\mathbf{W}_{out} = \mathbf{Y}_C \mathbf{X}_C^T \left(\mathbf{X}_C \mathbf{X}_C^T + \beta\,\mathbf{I}\right)^{-1}, \tag{3.12}$$

where,

$$\mathbf{X}_C = \left[\alpha \mathbf{X}_1,\, (1-\alpha)\,\mathbf{X}_2\right], \tag{3.13}$$

$$\mathbf{Y}_C = \left[\alpha \mathbf{Y}_1,\, (1-\alpha)\,\mathbf{Y}_2\right], \tag{3.14}$$

and,

$$\mathbf{X}_1 = \begin{bmatrix} \boldsymbol{q}(\boldsymbol{r}_1[t_l]) & \boldsymbol{q}(\boldsymbol{r}_1[t_l+1]) & \cdots & \boldsymbol{q}(\boldsymbol{r}_1[t_t]) \end{bmatrix}, \tag{3.15}$$

$$\mathbf{Y}_1 = \begin{bmatrix} \boldsymbol{u}_1[t_l] & \boldsymbol{u}_1[t_l+1] & \cdots & \boldsymbol{u}_1[t_t] \end{bmatrix}, \tag{3.16}$$

and similarly for $\mathbf{X}_2$ and $\mathbf{Y}_2$. $\mathbf{I}$ is the identity matrix of the appropriate size.

While the blending technique and training method described above focuses on using a RC to reconstruct a coexistence of only two attractors, it is scalable to include further attractors by using a different approach to the blending technique or by applying the same weight to all attractors during the training.

3.2.3 Using the multifunctional RC

When the training is complete, the predicting RC is described by the same autonomous continuous-time dynamical system as in Eq. (2.19). If the RC achieves multifunctionality, then according to Definition 3.2.1, for each $\mathcal{A}_1$ and $\mathcal{A}_2$ $\exists$ two attractors $\mathcal{S}_1, \mathcal{S}_2 \in \mathbb{S}$. When Eq. 2.19 is initialised with either $\hat{\boldsymbol{r}}(0) = \boldsymbol{r}_1(t_{\text{train}})$ or $\hat{\boldsymbol{r}}(0) = \boldsymbol{r}_2(t_{\text{train}})$ and the state of the multifunctional RC is projected from $\mathbb{S}$ to $\mathbb{P}$ using $\hat{\psi}$, the long-term dynamics of the reconstructed attractors, $\hat{\mathcal{A}}_1$ and $\hat{\mathcal{A}}_2$, will resemble the long-term dynamics of $\mathcal{A}_1$ and $\mathcal{A}_2$. A schematic of a multifunctional RC in operation is provided in Fig. 3.3.

3.3 Related works

It is important to stress that when a RC achieves multifunctionality, through the method described in the previous section, the RC has learned how to facilitate the coexistence of more than one attractor in $\mathbb{S}$ in such a way that when these attractors are projected to $\mathbb{P}$ then the desired output patterns are produced. This is the core research contribution established in this Thesis.

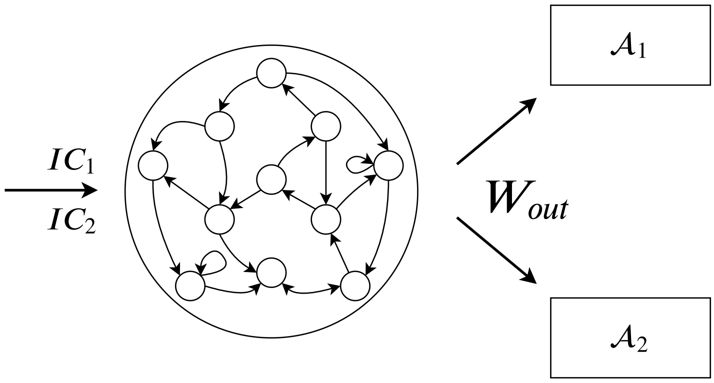


Figure 3.3: Illustration of the basic working features of a multifunctional RC. For each initial condition (IC), IC_1 and IC_2, the state of the multifunctional RC approaches its reconstruction of $\mathcal{A}_1$ and $\mathcal{A}_2$.

Nevertheless, there are many related works in the RC and wider ML community which consider training various ANNs designs to perform more than one task [109–116]. Be that as it may, unlike the results presented throughout this Thesis which focus on training RCs to achieve multifunctionality through exploiting multistable dynamics, multistability is not considered in the approaches cited above and is, in general, not widely explored in ML.

The works cited above can be divided into two categories, modular-weight and fixed-weight neural networks (MWNNs and FWNNs), and their relation to multifunctionality are discussed in the following Secs. 3.3.1 and 3.3.2.

3.3.1 Modular-weight neural networks

MWNNs are in general defined as networks which can either adapt their weights or have a library of pre-trained weights to choose from in order to solve a given problem and in doing so do not make use of multistability.

Systems within systems

Related to the concept of multifunctionality in ANNs is the notion of 'systems within systems' whereby distinct sub-networks are assigned particular duties within a larger network to collectively perform different behaviours. For example, the 'modular selection and identification control' technique proposed by Wolpert and Kawato in [109] focuses on a 'modular neural network' setup whereby a given trained network is singled out amongst others and brought into operation by a responsibility signal in order to generate the specific behaviour required from it.

Conceptors

Modular type network structures have also been put forward within the RC community, for instance, in the seminal work of Maass *et al.* [30] and Jaeger's work on 'conceptors' [111] similar principles are used to train RCs to perform different tasks. In these setups, for a given $\mathbf{M}$ and $\mathbf{W}_{in}$, a RC is trained on multiple tasks with a different $\mathbf{W}_{out}$ used to perform each corresponding task.

Adaptive $\mathbf{W}_{out}$

Furthermore, RCs have also been trained to switch between learning different chaotic and periodic attractors based on an external input like, for example, in the work of Inoue *et al.* in [112] and Lu and Bassett in [113]. In these scenarios, $\mathbf{W}_{out}$ is updated online in response to a given driving input from an individual attractor and only one attractor is shown to exist at each point in time.

3.3.2 Fixed-weight neural networks

The FWNNs discussed below, as the name suggests, have a fixed set of weights and the network performs different tasks through either a time-varying control input or change in some network parameter. However, in the works cited below, the FWNNs are never at any time multistable.

Parametric Biases

Tani *et al.* demonstrated in [110] that multiple temporal patterns can be learned using an RNN with a 'parametric biases' setup. Here the parametric biases act as bifurcation parameters which change the dynamical regime of the network in order to generate a specific behaviour.

Parameter-aware RC

The defining characteristic of the RC approach to ML is that to train the RC it is only necessary to determine a suitable readout layer, $\mathbf{W}_{out}$, to solve a given problem. However, since the first results regarding multifunctionality in a RC have been published in [31], it has recently been discovered that a given $\mathbf{W}_{out}$ can enable a RC to perform more than one task using different principles. In this context, RCs have been trained to interpolate dynamics and learn global bifurcation structures in 'parameter-aware' setups [114, 115], and also to anticipate synchronisation between systems [116].

In these RC designs an additional parameter input channel is incorporated into the RC's architecture during training such that the RC will perform a task associated with a given parameter value while $\mathbf{W}_{out}$ remains fixed after the training. What is more is that in these examples of parameter-aware RCs it is possible for the RC to go through bifurcations similar to the attractor from the dynamical system used to generate the training data without needing to be trained on data after the bifurcation takes place.

Adaptive controller

Fixed weight RCs have also been used in adaptive control problems. For instance, Oubbatti *et al.* demonstrate in [117] that a RC can be trained to act as a controller to supply various inputs to a given system in response to a changing environment. Several experiments have been conducted by Oubbatti *et al.* which highlight how a trained RC with fixed weights alters its state in order to provide an input to a system which, for example, enables the system to track reference signals constructed from randomised parameter settings of the system which are different to those experienced during the training in [118] and control the motion of multiple robots in [119].

Adaptive FWNNs

In the majority of the research cited in this section, a connection has been made with the works of Feldkamp *et al.* [120], Prokhorov *et al.* [121] and Santiago [122] on how fixed weight RNNs can produce adaptive behaviour and act as 'context discerning multifunction networks' respectively. These RNNs, in the form of recurrent multilayer perceptrons (RMLPs), are trained on time-series generated from a system whose parameters change randomly. After the training, the weights of the RMLPs remain fixed and the resulting network is referred to as a FWNN. By changing the input to this FWNN to correspond with changes in the parameters from the target system dynamics (which are different from those used during the training), it is demonstrated that the FWNN can successfully imitate the varying output patterns of the target system.

3.3.3 Flip-flop task

The work of Ceni *et al.* in [123] is the exception to the above works as the multistable dynamics of low-dimensional RCs ($N = 2$) are exploited to subsequently solve the 'flip-flop' task. In this scenario, the state of the RC must switch between different fixed points in response to a given input signal. An example of a higher dimensional input-driven RC ($N = 500$) which is specifically trained to solve the flip-flop task using a ridge regression procedure is also investigated. The task was first investigated by Sussillo and Barak in [38] when using a RC with $N = 1000$, the aim of the training is for the RC to learn how to interpret whether a given input signal requires the state of the RC to switch between different fixed points. However, in [123] and [38], it was not demonstrated if the RC can be trained to become multistable in the absence of the input signal during the training approaches used in each of these works.

3.3.4 Key differences to multifunctional RC

In contrast to the results cited in Secs. 3.3.1-3.3.3, the multifunctional RCs studied in this Thesis are not modular nor require a conscious change in parameters or inputs in order to solve different problems. When the training is successful, the multifunctional RC framework presented in this Thesis operates on a global scale where the artificial neurons exhibit multistable dynamics and perform multiple mutually exclusive tasks.

Although Santiago refers to this type of RNN behaviour as having a 'multifunction' in [122], there is no reference made to how this relates to multifunctionality in the context of biological neural networks or multistability. However, it is suggested by Santiago that ESNs may be a good candidate for producing FWNNs with multiple functions thereby recalling information from different memories. The significance of understanding how FWNNs accomplish this is also appreciated by Santiago and further theorised that this aspect of ANNs having multiple functions is necessary in the construction of machines that display more general intelligence.

Despite the differences between the works mentioned above on FWNNs in Sec. 3.3.2 and the work presented on multifunctional RCs in this Thesis, the question of, 'what is the behavioural capacity of RNNs?', also arises and serves as further motivation towards investigating the limits of what can be achieved with multifunctionality.

3.4 Advantages of multifunctionality

Aside from fusing ANNs with further characteristics present in their biological counterparts, translating multifunctionality to a machine learning environment opens the door for new application areas.

3.4.1 Data-driven modelling

Multistability is widely observed throughout nature and a prominent feature of many dynamical systems, in particular where transitions between coexisting attractors occur [105, 124, 125]. Despite the recent burst of interest in data-driven modelling of systems and processes which evolve over time, there is a lack of ML based solutions in modelling multistabilities in data and secondary phenomena related to multistability, for instance, transitions between attractors, attractor merging crises or bistable hysteresis cycles.

Multifunctionality extends the modelling capabilities of RCs by incorporating multistability into the domain of attractor reconstruction. This not only enables RCs to perform data-driven modelling of systems with multistability and address the current gap of multistability use-cases in ML but also to account for scenarios where multistability is suspected to play a role, for instance, in the modelling of transitions between attractors or states of operation. In the spirit of Barak's beliefs expressed in [126], that RNNs can serve as both hypothesis generation and testing tools, multifunctionality in the context of ML can be exploited to produce exotic cases of multistability that ordinarily no dynamical system can and subsequently allow for the generation and testing of hypothesis on how transitions between trivial and nontrivial coexistences of attractors occur.

While it is one thing to train an ANN to solve a single task, it may well be the case that the network architecture has the potential for a much broader functional capacity, hitherto underutilised, that can be unlocked to solve additional tasks concurrently. Networks which achieve multifunctionality can be seen as an efficient means of broadening the dynamical repertoire of artificially intelligent systems by taking greater advantage of the extent of their latent computational capabilities.

3.4.2 Memory Recall

Storing and retrieving memories

Several solutions to various machine learning problems can, in theory, coexist in a single multifunctional RC, whereby on demand of a given duty the corresponding learned memory, stored as an attractor in the RC's state space, is retrievable through changing the state of the RC via a suitably designed control command input or intrinsic task switching process.

Neuromodulators

The above discussion draws further parallels to biological neural networks, where such control mechanisms are comparable to neuromodulators [127], a hierarchical system of neurons which are believed to be involved in instigating the switching of activity patterns in multifunctional neural networks. Alternatively, as discussed in Sec. 3.3.2, it is also possible to alter the RC's state online by changing the RC's parameters or sets of learned weights, however, this reintroduces many issues which RCs were originally designed to avoid and multifunctionality naturally evades the catastrophic risks involved in crossing a 'tipping point' whereby phenomena such as points of no return or return tipping occur for certain parameter changes in nonautonomous systems [125, 128].

3.4.3 Computational Efficiency

In keeping with the theorised evolutionary origins of multifunctionality in BNNs as discussed in Sec. 3.1.1, training an ANN to become multifunctional and perform multiple tasks concurrently with a single network offers savings in both computational efficiency and memory usage. For instance, the examples of MWNNs discussed in Sec. 3.3.1 require either the storage of multiple sets of trained weights or the simulation of multiple task specific networks in order to switch between performing different tasks. Furthermore, as all memory storage and computational modes of operation are contained within the one multifunctional network, multifunctionality provides a scaling solution for novel computing schemes such as the physical RCs discussed in Sec. 2.3.5.

3.4.4 Chaotic Itinerancy

Multifunctionality also presents itself as a suitable means to realise Tsuda's thoughts on the role of 'chaotic itinerancy' in cognitive neurodynamics in [129]. Chaotic itinerancy describes the dynamical phenomenon of an autonomous switching process whereby the state of a given autonomous dynamical system switches between several 'attractor ruins' or 'quasi-attractors' which were previously stable and are now connected to form one global attractor. These quasi-attractors retain much of their original features except trajectories on these quasi-attractors leak into each other.

This type of behaviour is put forward by Tsuda as a potential dynamical mechanism to describe how the brain transitions between performing different tasks or recounting different memories characterised by each attractor ruin. Achieving multifunctionality in a RC can be used as a precursor to generate a specifically designed chaotically itinerant orbit between several attractor ruins. This connection between multifunctionality and chaotic itinerancy is first reported in Chapter 5 and used to model dynamical features of epilepsy in Chapter 8.

Chapter 4

Realising Multifunctionality

Considering the discussions from Chapters 2-3, that an ANN in the form of a RC can be trained to reconstruct the dynamics of a single attractor and that the behaviour of a given multifunctional BNN resembles a system with a coexistence of many attractors, in this chapter these phenomena are brought together in order for multifunctionality to be realised in a RC.

The ability of a RC to become multifunctional is highlighted through several numerical experiments, each of these tasks require the RC to reconstruct a coexistence of chaotic attractors in various circumstances. The dynamics of the multifunctional RCs are explored and the roles of UAs are assessed.

In each numerical experiment the target attractors are chosen such that there are no overlapping regions or intersecting trajectories between them in state space, this issue is considered later in Chapters 5-8.

4.1 Outline of Numerical Experiments

The numerical experiments which form the basis of this Chapter are now outlined. As motivated in Secs. 3.1.3 and 3.2.1, the aim of each experiment is to train a RC to become multifunctional by reconstructing a coexistence of attractors in $\mathbb{S}$ that resemble the target dynamics when projected to $\mathbb{P}$. As a source of training data, trajectories on chaotic attractors are generated from:

- Case I: a system which is already multistable.
- Case II: different parameter settings of the same system.
- Case III: two different systems entirely.

4.1.1 Choice of training data

A system which satisfies the aims of Case I and II was presented by Guan *et al.* in [130] and is described by the following set of equations,

$$\begin{aligned} \dot{x}_1(t) &= a\,x_1(t) - x_2(t)\,x_3(t) - x_2(t) + d, \\ \dot{x}_2(t) &= -b\,x_2(t) + x_1(t)\,x_3(t), \\ \dot{x}_3(t) &= -c\,x_3(t) + x_1(t)\,x_2(t). \end{aligned} \tag{4.1}$$

Here, $\boldsymbol{x}(t) = (x_1(t), x_2(t), x_3(t))^T$, defines the state of the system at a given time t. $a, b, c > 0$, and $d \in \mathbb{R}$ are the system parameters.

When setting $(a, b, c, d) = (5, 15, 3, 12)$ there is a coexistence of two single-scroll chaotic attractors as illustrated in Fig. 4.1. Initialising Eq. (4.1) with $\boldsymbol{x}^{\mathcal{A}_1}(0) = (1, 1, 1)^T$, results in the state of the system settling on the attractor $\mathcal{A}_1$, seen as the blue trajectory in Fig. 4.1. The other attractor, $\mathcal{A}_2$, indicated by the orange trajectory in Fig. 4.1, is arrived at once Eq. (4.1) is initialised from $\boldsymbol{x}^{\mathcal{A}_2}(0) = (1, 1, -1)^T$.

Case I

Trajectories on $\mathcal{A}_1$ and $\mathcal{A}_2$ are considered as the training data for Case I. It is important to note that these attractors are not symmetric to one another and therefore symmetry is not a factor in training the RC here. The issues regarding symmetry and multifunctionality are explored in Chapter 5.

Case II

In Case II the task for the RC is to reconstruct a coexistence of attractors from different parameter settings of the same system. The chosen attractors are $\mathcal{A}_2$ from Fig. 4.1 and a double-scroll chaotic attractor, $\mathcal{B}_1$, which is reached by initialising Eq. (4.1) with $\boldsymbol{x}^{\mathcal{B}_1}(0) = (1, 1, 1)^T$ and setting the system parameters as $(a, b, c, d) = (5, 8, 2, 2)$. A trajectory on $\mathcal{B}_1$ is shown in Fig. 4.2.

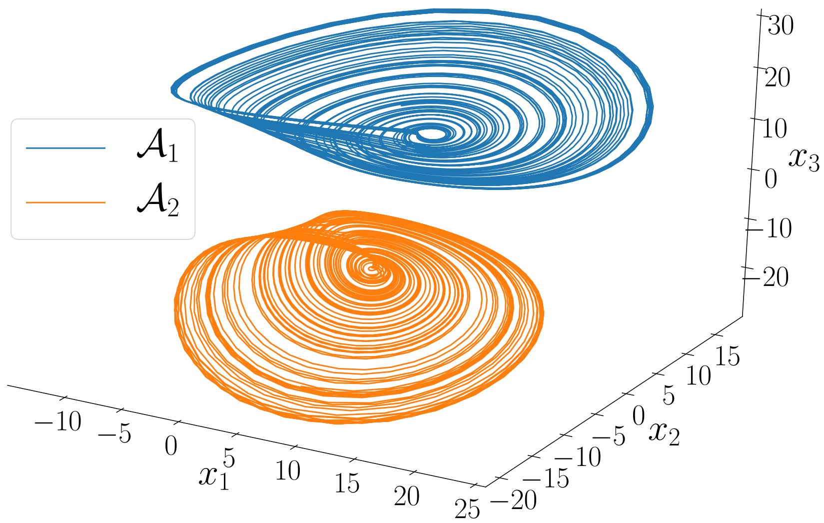


Figure 4.1: Coexisting attractors in the state space of Eq. (4.1) for $(a, b, c, d) = (5, 15, 3, 12)$ and $\boldsymbol{x}^{\mathcal{A}_1,\mathcal{A}_2}(0) = (1, 1, \pm 1)^T$.

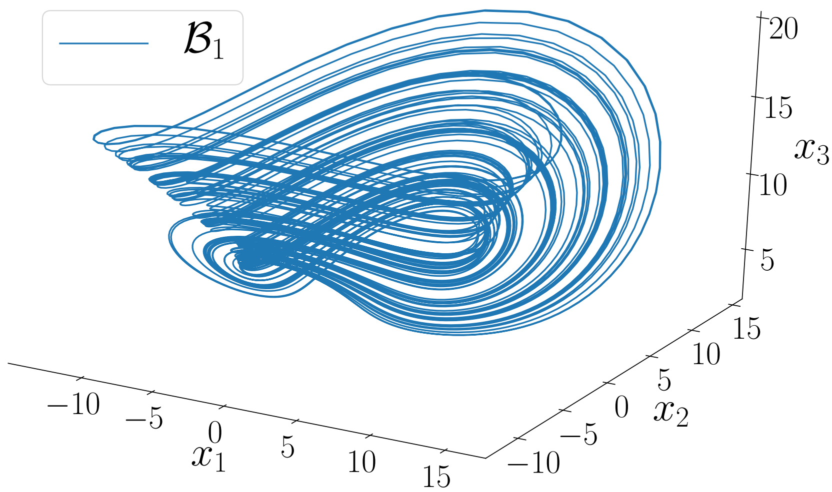


Figure 4.2: The attractor, $\mathcal{B}_1$, approached by the system in Eq. (4.1) when initialised from $\boldsymbol{x}^{\mathcal{B}_1}(0) = (1, 1, 1)^T$ with parameter values set to $(a, b, c, d) = (5, 8, 2, 2)$.

Case III

In Case III the RC is trained to reconstruct a coexistence of $\mathcal{A}_2$ and the chaotic butterfly-shaped attractor, $\mathcal{L}$, generated by the Lorenz system (defined in Eq. (2.28)). Trajectories on $\mathcal{L}$ are shown in previous figures throughout Chapter 2, see Fig. 2.3 for an example.

4.1.2 Training Data Precautions

It is important to highlight that $\mathcal{L}$, and the attractors, $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{B}_1$, generated by Eq. (4.1) share no common region in state space in each of the numerical experiments specified above. Otherwise, this adds a further level of difficulty in training a RC to distinguish which attractor a given trajectory belongs to.

In order to expose what criteria is needed for the RC to exhibit multifunctionality in the case of overlapping training data, the effects of decreasing the Euclidean distance between the attractors and target dynamics that the RC is trained to reconstruct a coexistence of are explored in Chapter 6.

4.1.3 Motivation behind experiments

While the dynamics of multifunctional BNNs are not described by trajectories on chaotic attractors or via some mathematical equation, the tasks considered in the present Chapter provide a reasonable analogy to the chaotic and highly variable behaviour which multifunctional BNNs have been reported to exhibit as discussed in Sec. 3.1.

At the same time, the coexistence of chaotic attractors is a naturally occurring phenomenon found in low-dimensional models of certain neural systems. For instance, in [131], a bifurcation analysis of a dynamical system describing a two neuron network with delay shows how two coexisting limit cycles undergo period doubling routes to chaos resulting in a coexistence of two single-scroll chaotic attractors which subsequently merge to form a double-scroll chaotic attractor. In various parameter settings of a three neuron Hopfield neural network studied by Bao *et al.* in [132], the network is capable of producing either a coexistence of multiple limit cycles or chaotic attractors, and, in some instances coexistences of multiple limit cycles and chaotic attractors. Therefore, by training a high-dimensional ANN in the form of a RC to acquire particular multistable dynamics in $\mathbb{S}$ in order to achieve multifunctionality, the results presented in this Chapter and throughout this Thesis are also of relevance in the context of studying the dynamics of systems in the form of neural networks.

Considering that the 'attractors' of multifunctional BNNs which are described in Sec. 3.1.2, such as those associated with producing the swimming or crawling motion of *Hirudo medicinalis*, are characterised by a set of corresponding time series data, then these can also be reconstructed in the state space of a multifunctional RC using the methods developed in this Thesis. Through creating a data-driven model of a given multifunctional BNN this in turn opens the door for new research avenues in neuroscience and enables one to make further advances in assessing the differences between how biological and artificial networks solve the same problem or exhibit similar dynamics. Furthermore, considering the work presented in [52] where Sussillo *et al.* demonstrate how RCs can be trained to act as brain-computer interfaces by, for example, translating neural data into the corresponding movement and positioning of a monkey's limbs, there is an opportunity for multifunctionality to be integrated with this approach. A brain-computer interface which incorporates a multifunctional RC provides new use cases of this technology in scenarios where the particular organism's brain activity relies on multifunctional BNNs to switch between different modes of operation. However, the above concepts related to multifunctional BNNs are not further investigated in this Thesis due to the lack of readily available experimental data but serve as directions for future work. In saying that, the above concepts are explored to some extent in Chapter 8 in order to model the dynamical features of transitions in epileptic brains using a multifunctional RC.

4.1.4 RC Training Specifications

The continuous-time RC setup described in Eq. (2.10) is studied throughout this Chapter and trained according to the steps outlined in Sec. 2.3.2 to re-

Case	N	P	α	ρ	σ	γ	β	t_{listen}	t_{train}
I	1000	0.04	$[0, 1]$	$[0.1, 1.1]$	0.2	10	10^{-2}	200	400
II	1000	0.04	$[0, 1]$	$[0.1, 1.1]$	0.4	10	10^{-2}	200	400
III	1000	0.04	$[0, 1]$	$[0.1, 1.1]$	0.2	10	10^{-2}	200	400

Table 4.1: RC design and training parameters used in each of the specified cases.

construct a single attractor and in Sec. 3.2.1 to achieve multifunctionality. The readout function, $\hat{\boldsymbol{\psi}}\,(\boldsymbol{r}(t))$, is designed as described in Eq. (2.12) using the squaring technique with $\boldsymbol{q}(\boldsymbol{r}(t))$ as defined in Eq. (2.13). Solutions of the RC are computed using the 4^{th} order Runge-Kutta method with the following integration time step, $\tau = 0.01$. The RC design and training parameters used to perform the experiments outlined in the present section are provided in Table 4.1. These parameters are chosen via a grid search method based on assessing the RC's long-term dynamics in order to provide some insight towards the relationship between multifunctionality and certain RC training and design parameters. A more precise understanding of these parameter relationships is established in Chapters 6 and 7. The parameters of particular interest in the present Chapter are the blending parameter, α, and the spectral radius of the RC's internal connection matrix, ρ. In each of the numerical experiments conducted in this Chapter, the same randomly initialised $\mathbf{M}$ and $\mathbf{W}_{in}$ matrices are used and $\mathbf{M}$ is rescaled accordingly to a specific ρ.

Throughout the remaining sections of the present Chapter, the events in which a RC was trained to successfully reconstruct the coexistence of attractors as specified in Case I, II and III are illustrated. There is a focus on establishing the optimal α with respect to ρ in order to determine the regions in which multifunctionality was achieved. Following these observations, the circumstances in which the reconstruction of a given attractor fails are examined and a number of UAs are found. The evolution of these UAs with respect to α and ρ are tracked and from this a number of 'behind-the-scenes' bifurcations are uncovered.

4.1.5 Error Analysis

As mentioned in the previous section, the long-term dynamics of the RC are analysed in order to determine whether multifunctionality was achieved or not by reconstructing the dynamics of both attractors in Case I-III.

In order to compare the performance of a multifunctional RC that reconstructs a coexistence of, for instance, two chaotic attractors, CA_1 and CA_2, to a RC that is trained to reconstruct only CA_1, the following notation is introduced in order to distinguish the multifunctional readout matrix, $\mathbf{W}_{out}^{CA_1,CA_2}$, from the 'task specific' readout matrix, $\mathbf{W}_{out}^{CA_1}$.

There are many means of assessing the accuracy of a predicted time series in comparison to the ground truth. In this Chapter, a Normalised Root Mean Square Error (NRMSE),

$$\theta_{\mathcal{S}}(\mathbf{W}_{out}^{\mathcal{S}}) = \frac{1}{D}\sum_{i=1}^{D}\theta_{\hat{u}_i}(\mathbf{W}_{out}^{\mathcal{S}}), \tag{4.2}$$

is used in order to provide an indication of whether the RC reconstructs a particular attractor, $\mathcal{S}$, using the readout matrix $\mathbf{W}_{out}^{\mathcal{S}}$ that is trained for a specific $\mathcal{S}$ or in the case of multifunctionality for a specific number of attractors, and the NRMSE of each individual state variable, $i = 1, 2, \ldots, D$ (where $D = 3$ in this Chapter), is calculated as,

$$\theta_{\hat{u}_i}(\mathbf{W}_{out}^{\mathcal{S}}) = \frac{\sqrt{\frac{1}{t^*}\int_{t=t_{\text{predict}}-t^*}^{t_{\text{predict}}}\left(u_i(t) - \hat{u}_i(t)\right)^2 dt}}{|\max_t(u_i(t)) - \min_t(u_i(t))|}, \tag{4.3}$$

where, $\hat{u}_i(t)$, is the RC's prediction of $u_i(t)$ for a given state variable i at time t. For computational purposes, the integral in Eq. (4.3) is converted to a sum over j where, for instance, $u_i(t)$ is evaluated at discrete times $t_j = j + (t_{\text{predict}} - t^*)/\tau$ for $j \in (0, 1, 2, \ldots, \lfloor t^*/\tau \rfloor)$, where $\lfloor x \rfloor$ is the 'floor function' which returns the greatest integer $\leq x$ for $x \in \mathbb{R}$. The prediction end time, t_{predict}, is fixed as $t_{\text{predict}} = 600$ for each result presented in this Chapter and, $t_{\text{predict}} - t^*$ is time that error sampling begins from. t_{predict} is chosen to be sufficiently large in order for $\theta_{\mathcal{S}}(\mathbf{W}_{out}^{\mathcal{S}})$ to be evaluated from a trajectory on the particular reconstructed attractor that the predicting RC's state approaches. The max $(\cdot)$ and min $(\cdot)$ functions are measures of the maximum and minimum value of the time series evaluated from $t_{\text{predict}} - t^*$ to t_{predict}. The closer $\theta_{\mathcal{S}}(\mathbf{W}_{out}^{\mathcal{S}})$ is to 0 the more accurate the prediction of $\mathcal{S}$. It was found empirically that, in general, attractor reconstruction fails for $\theta_{\mathcal{S}}(\mathbf{W}_{out}^{\mathcal{S}}) > \delta = 0.35$.

While the above approach provides a reasonable indication on whether the predicting RC reconstructs a given attractor, it does not guarantee if attractor reconstruction was achieved. There are more robust methods like comparing the predicting RC's Lyapunov exponents on a given attractor to the dynamical system which generated the training data in the first place or computing the transversal Lyapunov exponents like in [59]. However, such an approach is significantly more computationally expensive to employ than assessing the error on the resulting predicted time series data with the approach outlined above. Furthermore, since the RC training and design parameters are chosen via a grid search method, in order to provide some intuition behind the necessary ingredients for a RC to achieve this new concept of multifunctionality, a stability study is not as feasible to conduct on a mass basis due to the high-dimensionality of the RC and the density of the resulting $\mathbf{W}_{out}$.

4.1.6 Reconstructing Individual Attractors

In order to determine if the attractors can be reconstructed in the usual sense from the method described in Sec. 2.4, an error analysis of the predicted time series is conducted when using the 'task specific' readout matrices, $\mathbf{W}_{out}^{\mathcal{A}_1}$, $\mathbf{W}_{out}^{\mathcal{A}_2}$, $\mathbf{W}_{out}^{\mathcal{B}_1}$, and $\mathbf{W}_{out}^{\mathcal{L}}$. In Fig. 4.3 the error analysis (with $t^* = 40$) indicates attractor reconstruction was achieved for all $\rho \in [0.1, 1.1]$ using each of the task specific matrices as $\theta_{\mathcal{S}}(\mathbf{W}_{out}^{\mathcal{S}}) < \delta$ in Case I-III.

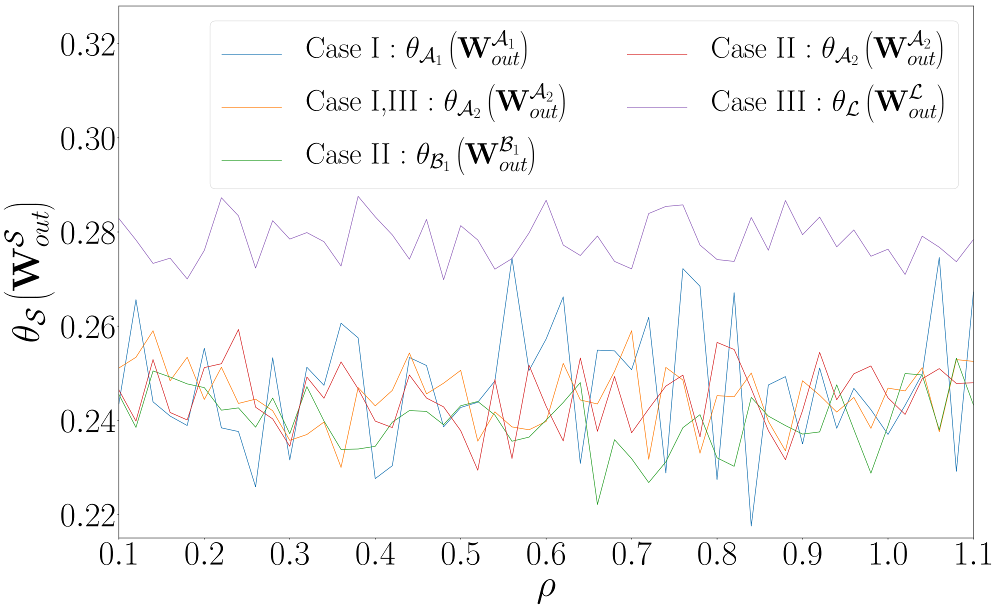


Figure 4.3: $\theta_{\mathcal{S}}(\mathbf{W}_{out}^{\mathcal{S}})$ vs. ρ when using the task specific matrices to reconstruct a given attractor $\mathcal{S}$ from Case I, II and III.

4.2 Reconstructing a Coexistence of Attractors

The next step is to search for values of α which give rise to a multifunctional readout matrix defined as $\mathbf{W}_{out}^{\alpha} = \mathbf{W}_{out}^{CA_1,CA_2}$ using the convention taken in the previous section, that allows the RC to reconstruct a coexistence of CA_1 and CA_2, or in the case of the current series of numerical experiments, a coexistence of the attractors specified in Case I-III. After applying the blending technique described in Sec. 3.2.2, the following error measure, $\varepsilon_{\mathcal{S}}\left(\alpha\right)$, defined as,

$$\varepsilon_{\mathcal{S}}\left(\alpha\right) = \theta_{\mathcal{S}}\left(\mathbf{W}_{out}^{\alpha}\right)/\theta_{\mathcal{S}}\left(\mathbf{W}_{out}^{\mathcal{S}}\right). \tag{4.4}$$

provides an indication on whether $\mathbf{W}_{out}^{\alpha}$ or $\mathbf{W}_{out}^{\mathcal{S}}$ results in the associated RCs producing a more accurate reconstruction of a given attractor, $\mathcal{S}$, according to $\theta_{\mathcal{S}}\left(\cdot\right)$ as if $\varepsilon_{\mathcal{S}}\left(\alpha\right) < 1$, then the prediction of $\mathcal{S}$ was more accurate when using $\mathbf{W}_{out}^{\alpha}$ over $\mathbf{W}_{out}^{\mathcal{S}}$ with the opposite being said if $\varepsilon_{\mathcal{S}}\left(\alpha\right) > 1$.

4.2.1 Case I

Starting with the task of reconstructing attractors from a multistable system in Case I for $\rho = 0.7$, the resulting $\varepsilon_{\mathcal{S}}(\alpha)$ vs. α plot is shown in Fig. 4.4a. Here it can be seen that, on one hand, as α is increased from 0, there is large decrease in $\varepsilon_{\mathcal{A}_1}$ and it stays relatively close to 1 for $\alpha \gtrsim 0.22$, while on the other hand, $\varepsilon_{\mathcal{A}_2}$ remains relatively near 1 for $\alpha \lesssim 0.88$, and then grows as α is increased thereafter. From this it can be deduced that multifunctionality is achieved by the RC for $0.22 \lesssim \alpha \lesssim 0.88$.

Figs. 4.4b-4.4c illustrate a successful implementation of the blending technique for $\alpha = 0.5$. The predicted trajectory on $\mathcal{A}_1$ and $\mathcal{A}_2$ when using the task specific matrices, $\mathbf{W}_{out}^{\mathcal{A}_1}$ and $\mathbf{W}_{out}^{\mathcal{A}_2}$, are plotted in orange and the multifunctional readout matrix, $\mathbf{W}_{out}^{\alpha} = \mathbf{W}_{out}^{\mathcal{A}_1,\mathcal{A}_2}$, are plotted in green. A trajectory on the actual attractors is plotted in blue in these figures.

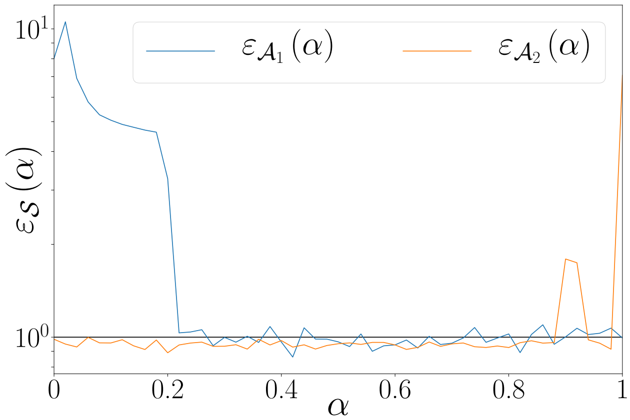
4.2.2 Case II - III

A similar analysis for Case II and III is conducted with Fig. 4.5 illustrating examples where the RC was successfully trained to be multifunctional. Fig. 4.5a shows that when training the RC to reconstruct both $\mathcal{B}_1$ and $\mathcal{A}_2$ in Case II, where $\mathbf{W}_{out}^{\alpha} = \mathbf{W}_{out}^{\mathcal{B}_1,\mathcal{A}_2}$, the desired coexistence of chaotic attractors is achieved for $\rho = 0.3$ and $\alpha = 0.5$. For Case III, where $\mathbf{W}_{out}^{\alpha} = \mathbf{W}_{out}^{\mathcal{L},\mathcal{A}_2}$, when setting $\rho = 0.85$ the RC is successfully trained to express multifunctionality as it can reconstruct both $\mathcal{L}$ and $\mathcal{A}_2$ for $\alpha = 0.65$ as seen in Fig. 4.5b.

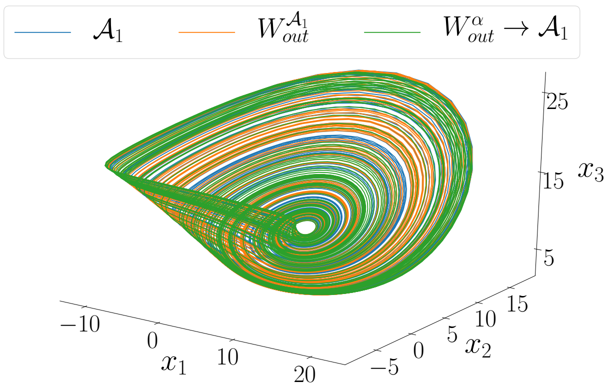
4.2.3 Additional Comments

Impact of results

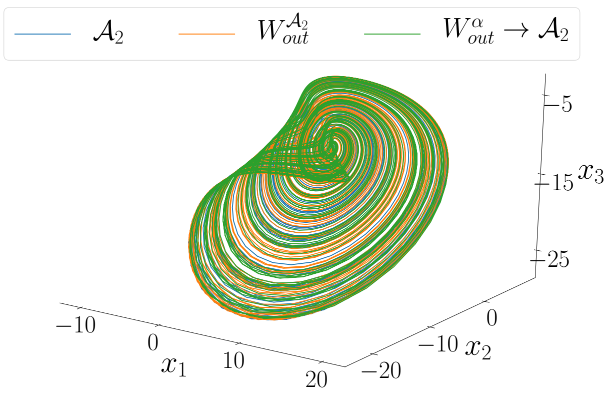
The results illustrated in Figs. 4.4-4.5 show that a RC can indeed be trained to exhibit multifunctionality which broadens the current set of applications that a RC is capable of. These results establish that, for instance, instead of having to change parameters in a system for it to exhibit a different behaviour, separate modes of operation from various parameter choices of the system can



(a) $\varepsilon_{\mathcal{S}}(\alpha)$ vs. α

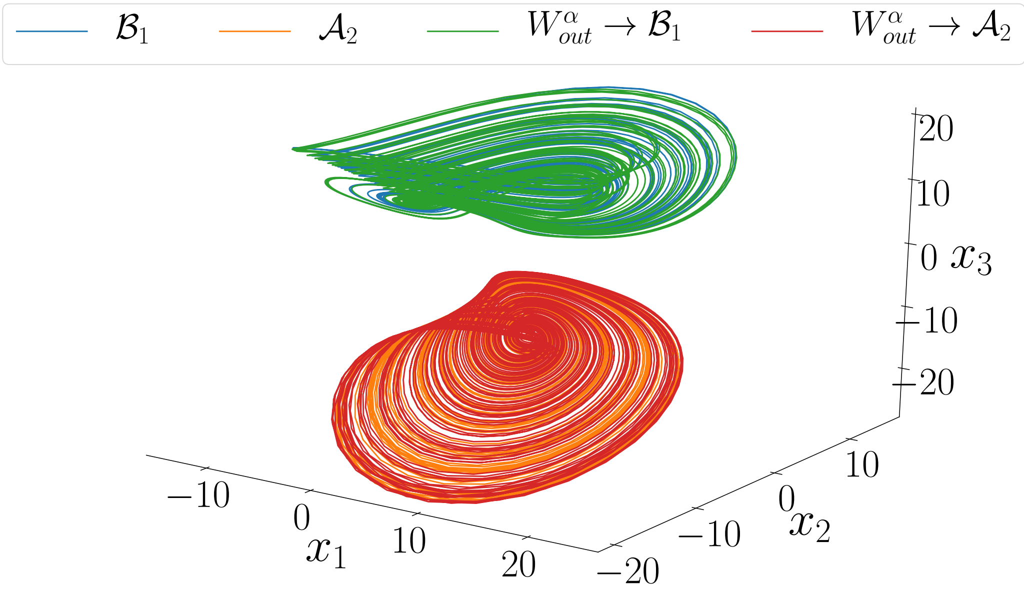


(b) Reconstruction of $\mathcal{A}_1$

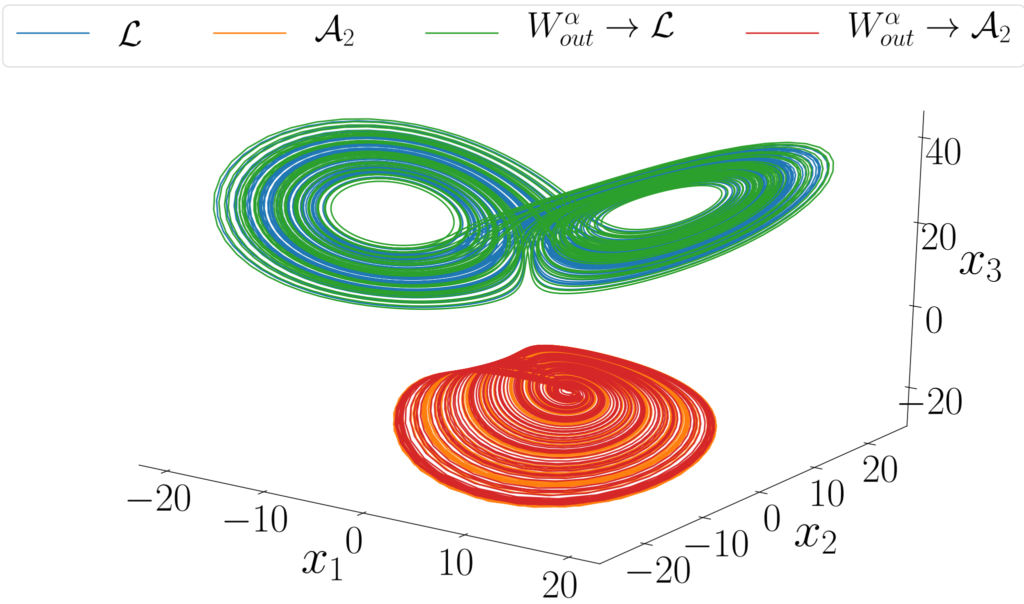


(c) Reconstruction of $\mathcal{A}_2$

Figure 4.4: Case I ($\rho = 0.7$) ; (a): $\varepsilon_{\mathcal{A}_1}$ and $\varepsilon_{\mathcal{A}_2}$ vs. α. (b)-(c): Attractor reconstruction using, $\mathbf{W}_{out}^{\mathcal{A}_1}$ and $\mathbf{W}_{out}^{\mathcal{A}_2}$ and $\mathbf{W}_{out}^{\alpha} = \mathbf{W}_{out}^{\mathcal{A}_1,\mathcal{A}_2}$ for $\alpha = 0.5$.



(a) Case II: Reconstructing attractors $\mathcal{B}_1$ and $\mathcal{A}_2$.



(b) Case III: Reconstructing the chaotic Lorenz attractor $\mathcal{L}$ and $\mathcal{A}_2$.

Figure 4.5: Illustration of attractor reconstruction in; (a) Case II for $(\alpha, \rho) = (0.5, 0.3)$ where $\mathbf{W}_{out}^{\alpha} = \mathbf{W}_{out}^{\mathcal{B}_1,\mathcal{A}_2}$ and (b) Case III for $(\alpha, \rho) = (0.65, 0.85)$ where $\mathbf{W}_{out}^{\alpha} = \mathbf{W}_{out}^{\mathcal{L},\mathcal{A}_2}$.

coexist in $\mathbb{P}$. In addition, this combination of attractors is not limited to a single system as it is possible to combine attractors from different systems to coexist in one. Given this ability, there is the prospect of designing an appropriate controller for Eq. (2.19) to switch between attractors. For further reading on 'inter-attractor control' see [133].

4.2.4 Parameter dependencies

Although Figs. 4.4-4.5 illustrate instances in which appropriate values of α were found to give rise to multifunctionality, this cannot be said for all α values nor for other choices of γ, ρ, σ and β. Moreover, relying on an error analysis alone is not sufficient enough to understand the dynamics of a multifunctional RC or to classify the parameter regions in which multifunctionality is achieved. As the RC used throughout this Chapter is designed such that if attractor reconstruction fails then the predicted trajectory will not blow up to infinity in a finite amount of time, however, the prediction can decay toward some other stable attractor. Furthermore, given the nature of these numerical experiments, it is also possible that the RC's predicted trajectory on one chaotic attractor can switch to the other chaotic attractor. Following this argument, in the next section a means of characterising the attractor that the prediction settles to is employed and from this the regions of multifunctionality in the (α, ρ)-plane are identified.

4.3 Regions of Multifunctionality

In this section the long term behaviour of the RC in the prediction stage (described by Eq. (2.19)) for each Case I-III is analysed.

4.3.1 Characterising Attractors

When the RC is trained for a given $\alpha \in [0, 1]$ and $\rho \in [0.1, 1.1]$, and subsequently initialised with $\hat{\boldsymbol{r}}(0)$ corresponding to $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{B}_1$ or $\mathcal{L}$ for the appropriate Case, the particular attractor that the state of the RC subsequently settles on is classified. A colour is assigned to each point in the $(\alpha,\, \rho)$-plane that characterises the prediction of a given attractor for $t_{\text{predict}} = 600$ according to the following criteria:

1. Yellow, if the predicted time series is periodic[1] for $t_{\text{predict}} - 40 \leq t \leq t_{\text{predict}}$, then the prediction has decayed to some limit cycle, LC.

2. Green, if the predicted time series remains constant for $t_{\text{predict}} - 10 \leq t \leq t_{\text{predict}}$, then the prediction has decayed to some fixed point, FP.

3. Magenta, if for $t^* = 40$, $\theta_{\mathcal{S}}\left(\mathbf{W}_{out}^{\alpha}\right) \leq \delta$ for the predicted time series in comparison to the target time series of the other chaotic attractor, then the prediction has switched from one chaotic attractor to the other.

4. Blue, if conditions 1-3 are not fulfilled and if for $t^* = 40$ that $\theta_{\mathcal{S}}\left(\mathbf{W}_{out}^{\alpha}\right) \leq \delta$ for a given attractor $\mathcal{S}$, then the dynamics of $\mathcal{S}$ was well reconstructed.

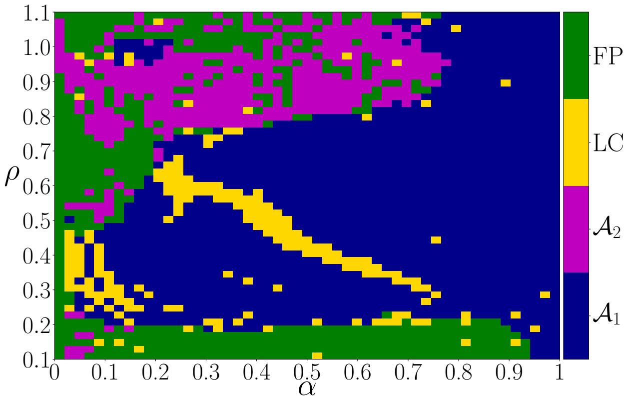
5. Red, if otherwise, in order to signal that closer inspection of the prediction is needed.

In the interest of exploring the relationship between multifunctionality and certain RC design and training parameters, the above criteria is used to provide a rough picture on the dynamics of a given multifunctional RC. More robust criteria would, for instance, involve computing the transversal Lyapunov exponents to ensure the existence of stable response attractors in $\mathbb{S}$ for a given task however this is significantly more computationally expensive than the proposed criteria above.

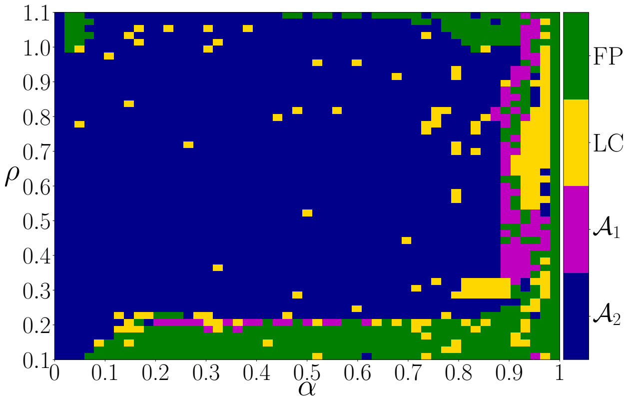
Characterising Attractors for Case I

The result of applying the above prediction analysis for Case I is shown in Fig. 4.6. It can be seen here that attractor reconstruction failed when initialising the predicting RC with $\hat{\boldsymbol{r}}(0)$ corresponding to $\mathcal{A}_1$ in Fig. 4.6a ($\mathcal{A}_2$ in

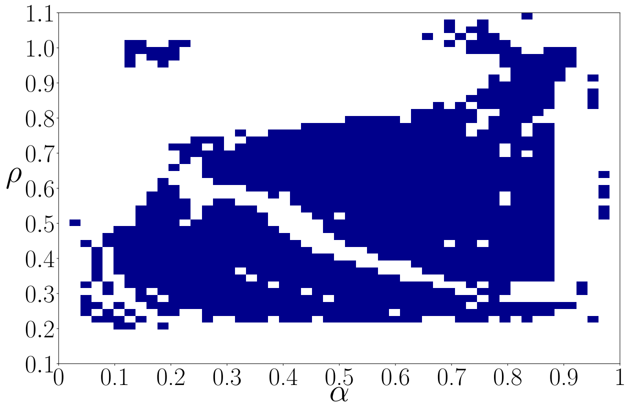
[1]If the location of all the maximum of the local maxima (up to a relatively small difference) of the time series of a given dynamical variable repeats every k steps, then the time series is considered to be periodic. This convention is used throughout this Thesis to assess if a given time series is periodic.



(a) IC: $\mathcal{A}_1$

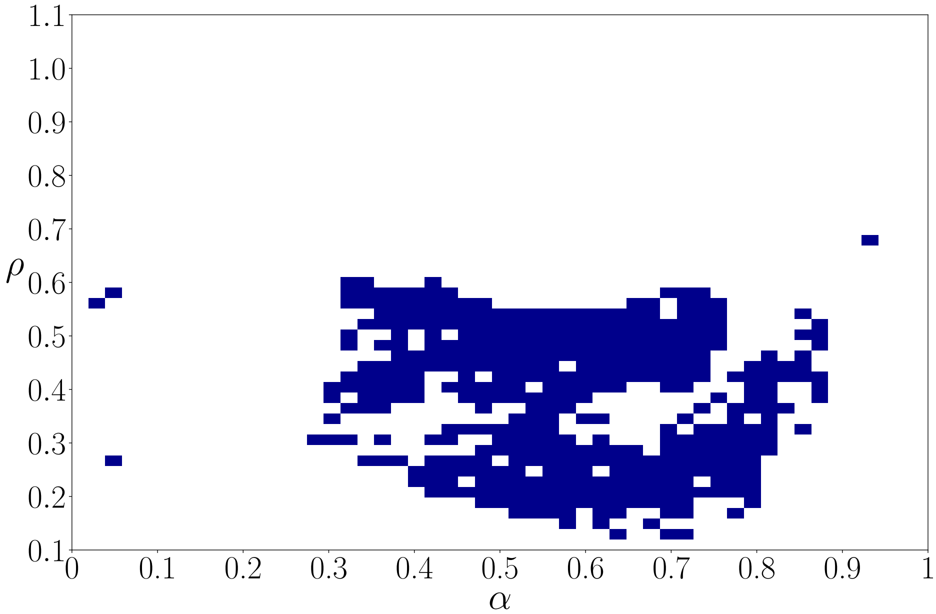


(b) IC: $\mathcal{A}_2$

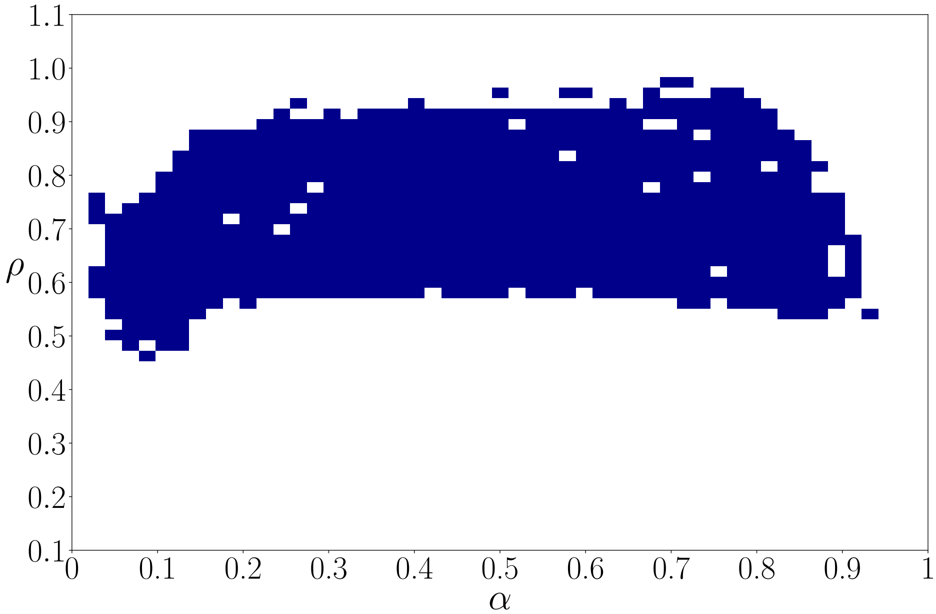


(c) Regions of multifunctionality

Figure 4.6: (a)-(b):Long-term behaviour of prediction in the (α, ρ)-plane for Case I. Each colour characterises the attractor the RC settles to from a particular IC: Initial Condition. (c): Plotted in blue are the regions of multifunctionality.



(a) Case II



(b) Case III

Figure 4.7: Regions of multifunctionality in the (α, ρ)-plane for; (a) Case II and (b) Case III.

Fig. 4.6b) for $\alpha = 0$ (1). This is in contrast to the recent findings of Rohm *et al.* in [102] which show that when a RC is trained to reconstruct only one attractor from a multistable system it simultaneously inherits the ability to reconstruct all the other attractors and provide a coexistence of these in $\mathbb{P}$.

As depicted in Figs. 4.6a-4.6b, when $\alpha = 0$ or 1, the prediction consistently decays towards some fixed point for all values of ρ. However, when increasing (decreasing) α from 0 (1) there is a more varied sequence of events for a given ρ. Prior to achieving attractor reconstruction, the predicted trajectory on $\mathcal{A}_1$ ($\mathcal{A}_2$) at times tends towards some limit cycle or switches to $\mathcal{A}_2$ ($\mathcal{A}_1$). Fig. 4.6a also shows that for large ρ, the RC favours the reconstruction of $\mathcal{A}_2$ as opposed to $\mathcal{A}_1$ where the prediction of $\mathcal{A}_1$ mainly switches to $\mathcal{A}_2$ until some critical α value where attractor reconstruction is achieved.

Furthermore, as there are no red points found to occur in Figs. 4.6a-4.6b, this on one hand indicates that closer inspection of the RC's dynamics is not required to assess its long term dynamics and on the other showcases the robustness of the prediction assessment criteria in successfully characterising the dynamics of the multifunctional RC.

4.3.2 Multifunctionality in the (α, ρ)-plane

Nevertheless, there is some middle ground where both attractors are successfully reconstructed for a given pair of α and ρ, this common blue area between Figs. 4.6a-4.6b are the regions in which multifunctionality was achieved. A separate picture in Fig. 4.6c is provided to show these regions explicitly. Through the same reasoning in Fig. 4.7a and Fig. 4.7b the regions in the (α, ρ)-plane in which the RC was successfully trained to express multifunctional behaviour are shown for both Case II and Case III.

There is a relatively large common blue region in the $(\alpha,\ \rho)$-plane for Case I in Fig. 4.6c. However, for Case II and III the regions of multifunctionality in Figs. 4.7a-4.7b are relatively much smaller. Case II and III require a higher level of dynamical flexibility from the RC as not only do the chaotic attractors come from different settings of Eq. (4.1) and different systems entirely but also vary characteristically, i.e. single-scroll and double-scroll. At the same time, a particular setup or random realisation of the RC may happen to favour reconstructing one flavour of attractor over the other, thus a contributing factor towards the reduced regions of multifunctionality seen here.

4.3.3 Issues with timescale

While in Case I-III the target chaotic attractors are separated in state space, they do share some dynamical similarities which may be just as vital in order to achieve multifunctionality. For example, the chosen attractors in all three of the studied cases evolve along a similar timescale. Using the pair of chaotic attractors in Case I as an example, the relationship between the RC's capacity to exhibit multifunctionality when changing the timescale of $\mathcal{A}_1$ while keeping the timescale of $\mathcal{A}_2$ fixed is investigated. To do this, a new timescale parameter is introduced, $\Delta^{\mathcal{A}_1}$, which is used to control the timescale of $\mathcal{A}_1$. The RHS of Eq. (4.1) is multiplied by $\Delta^{\mathcal{A}_1}$ and the system is initialised with $\boldsymbol{x}^{\mathcal{A}_1}(0) = (1, 1, 1)^T$ to generate solutions of $\mathcal{A}_1$ with a modified timescale. If $\Delta^{\mathcal{A}_1} < 1$ then the dynamics are slowed down and if $\Delta^{\mathcal{A}_1} > 1$ the dynamics are sped up. An example of the effect this has on the dynamics of the training data is provided in Fig. 4.8 for $\Delta^{\mathcal{A}_1} = 0.6, 1.0$, and 1.4.

The regions of multifunctionality in the $(\Delta^{\mathcal{A}_1}, \rho)$-plane for $\alpha = 0.5$ are identified using the same process which produced the results shown previously in Figs. 4.6c and 4.7. The result of this is displayed in Fig. 4.9 which shows that if $\mathcal{A}_1$ evolves along a relatively longer or shorter timescale in comparison to $\mathcal{A}_2$, then there is a point where multifunctionality is lost for all ρ values. This reveals the limits upon which multifunctionality can be achieved in this scenario by exposing the RC's dynamical capacity to facilitate the coexistence of increasingly dissimilar chaotic attractors in $\mathbb{P}$.

4.3.4 Dynamics prior to reconstruction

Now, returning to the results shown in Fig. 4.6a, where the prediction of $\mathcal{A}_1$ for $\rho = 0.7$ as α is increased from 0, before multifunctionality is achieved the prediction repeatedly falls to a fixed point. These fixed points for various val-

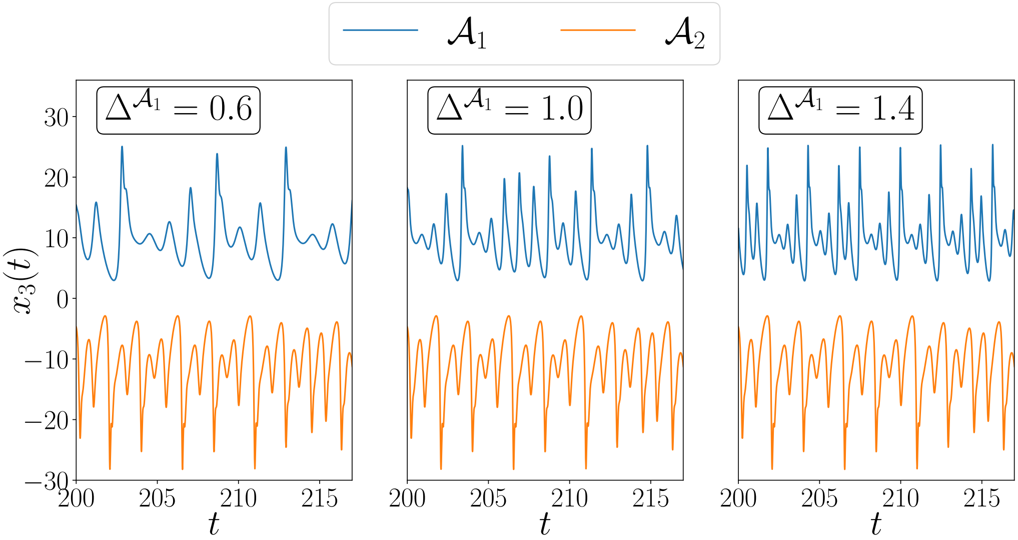


Figure 4.8: Illustrating the influence of different choices of $\Delta^{\mathcal{A}_1}$ on the dynamics of $\mathcal{A}_1$ in comparison to $\mathcal{A}_2$.

ues of α are plotted in Fig. 4.10. Also seen here is a case where the prediction switches from $\mathcal{A}_1$ to $\mathcal{A}_2$ when $\alpha = 0.2$.

In light of the results above the following question arises, what happens to these fixed points in Fig. 4.10 as multifunctionality is achieved? Is it the case that the successful reconstruction of $\mathcal{A}_1$ entirely takes the place of these fixed points in $\mathbb{P}$ or, do these fixed points still exist except they cannot be seen from looking directly at the reconstructed attractor? Furthermore, are there other attractors lurking in $\mathbb{P}$? These questions are explored in the following section and the important role of ICs in the prediction stage of the RC are highlighted.

4.4 Detecting Untrained Attractors

4.4.1 Probing $\mathbb{P}$

In order to get a broader picture of $\mathbb{P}$ for a given ρ and α in Case I, the same approach from Sec. 2.4 is used in the present section, the trained RC is initialised with 1000 random ICs and its resultant dynamics are examined. A

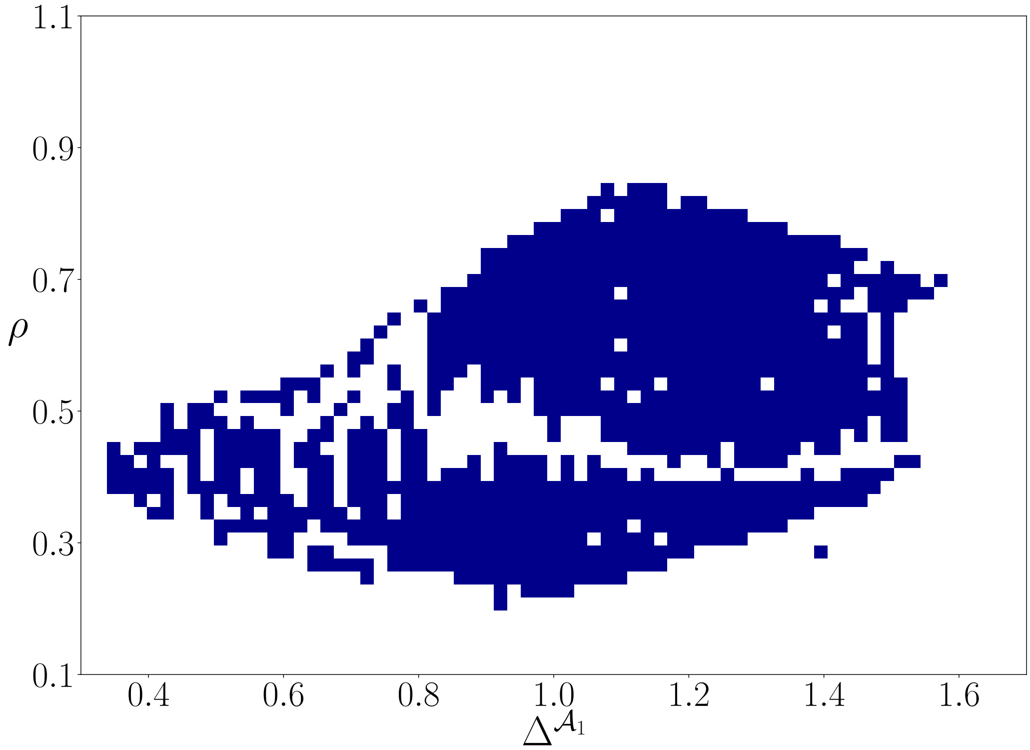


Figure 4.9: Regions of multifunctionality in the $(\Delta^{\mathcal{A}_1}, \rho)$-plane when changing the timescale of $\mathcal{A}_1$ in Case I with timescale parameter $\Delta^{\mathcal{A}_1}$.

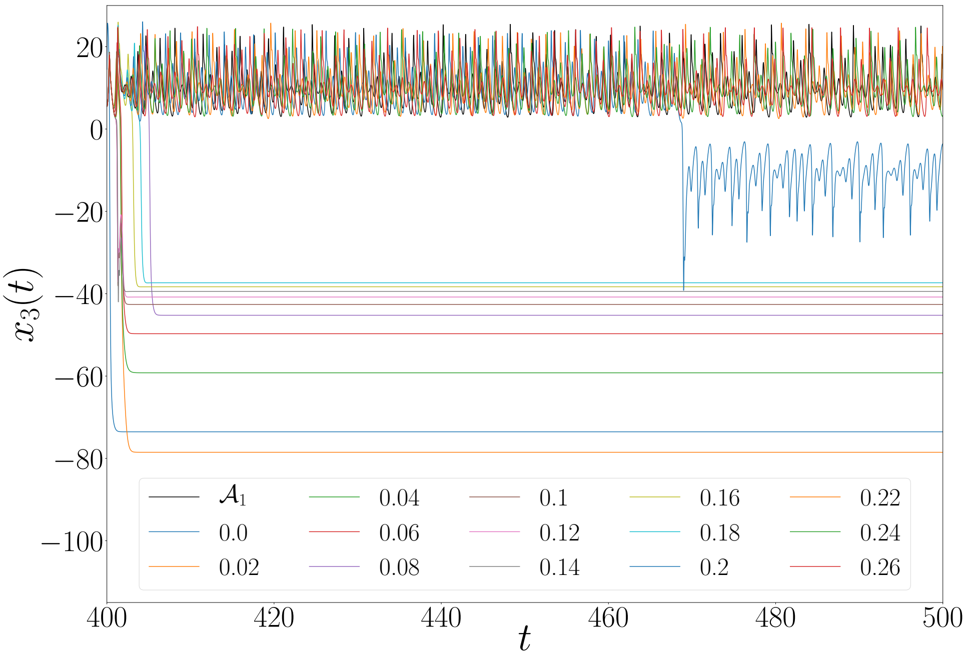


Figure 4.10: Case I ($\rho = 0.7$, IC: $\mathcal{A}_1$): $x_3(t)$ vs. t for the reconstruction of $\mathcal{A}_1$ (coloured in black) and the predicted trajectories for various values of α as indicated in the plot legend.

result of this is shown in Fig. 4.11. Plotted here are a number of trajectories, $x_3^{\boldsymbol{\xi}_i}$, which describe the time-evolution of the $x_3(t)$ variable as generated by the RC starting from the i^{th} random IC, $\boldsymbol{r}^{\boldsymbol{\xi}_i}(0)$, when $\rho = 0.7$ and the RC is trained for $\alpha = 0.45$, 0.50, and 0.55. For $\alpha = 0.5$ there exists a stable fixed point located within the middle of both reconstructed chaotic attractors. As more weight is applied to the data from $\mathcal{A}_2$ with $\alpha = 0.45$, another stable fixed point appears closer to $\hat{\mathcal{A}}_2$. Similarly for $\alpha = 0.55$ a stable fixed point closer to $\hat{\mathcal{A}}_1$ is found. So while the RC was successfully trained to reconstruct a coexistence of $\mathcal{A}_1$ and $\mathcal{A}_2$ there are several UAs populating $\mathbb{P}$ that have no connection to the dynamical system described in Eq. (4.1).

4.4.2 Considering ρ and α as bifurcation parameters

The results in Fig. 4.11 show that small changes in α can give rise to a bistability of UAs. This suggests that there are some 'behind-the-scenes' bifurcations taking place in $\mathbb{P}$. However, further discussion is needed in order to regard α as a bifurcation parameter of the RC. While ρ is a parameter of the RC itself, α is a parameter that is strictly involved in the training procedure. Therefore to consider α as a bifurcation parameter of the RC it is necessary to assess if, for a small change in α there is a relatively smooth change in the elements of $\mathbf{W}_{out}^{\alpha} = \mathbf{W}_{out}^{\mathcal{A}_1,\mathcal{A}_2}$ generated from a specific combination of α and ρ. In other words, a continuous function which maps $(\alpha,\, \rho) \to \mathbf{W}_{out}^{\alpha}$ needs to exist.

Assessing the relationship between $\boldsymbol{\alpha}$, $\boldsymbol{\rho}$, and $\mathbf{W}_{out}^{\boldsymbol{\alpha}} = \mathbf{W}_{out}^{\mathcal{A}_1,\mathcal{A}_2}$

In order to expand upon the above notion, elements of $\mathbf{W}_{out}^{\alpha} = \mathbf{W}_{out}^{\mathcal{A}_1,\mathcal{A}_2}$ are chosen at random and their relationship with respect to changes in α and ρ is examined. The result of this is shown in Fig. 4.12 for four randomly chosen elements of $\mathbf{W}_{out}^{\alpha}$.

The smoothness of the surface plots in Fig. 4.12 indicate that a relatively small change in α or ρ in turn gives rise to a small change in the elements of $\mathbf{W}_{out}^{\alpha}$, to which in a generalised sense, contributes to an overall change in the dynamics of the RC. From this result, α is now considered as a bifurcation parameter of the RC. The blending technique and the training procedure can

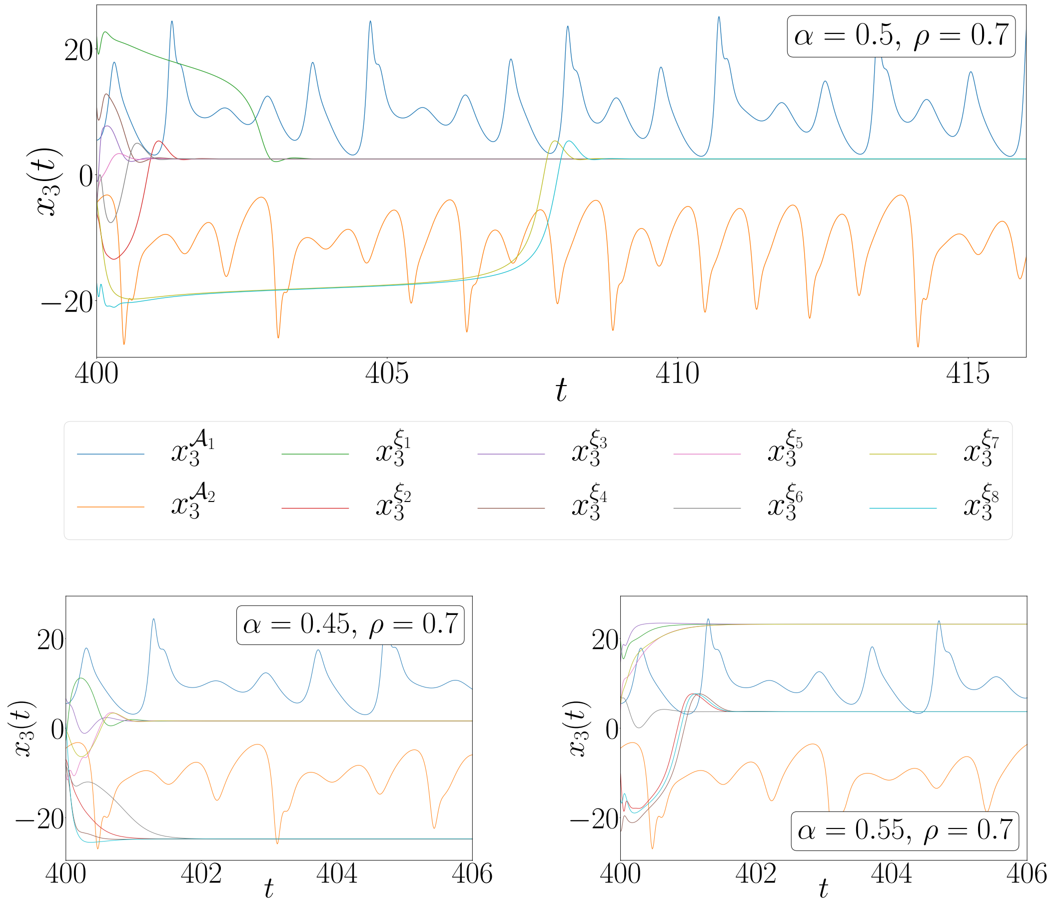


Figure 4.11: Time trace of reconstructed $x_3(t)$ variable for RC trained with α and ρ as indicated in each plot and initialised with a given random IC, $\boldsymbol{\xi}_i$ for $i = 1, 2, \ldots$.

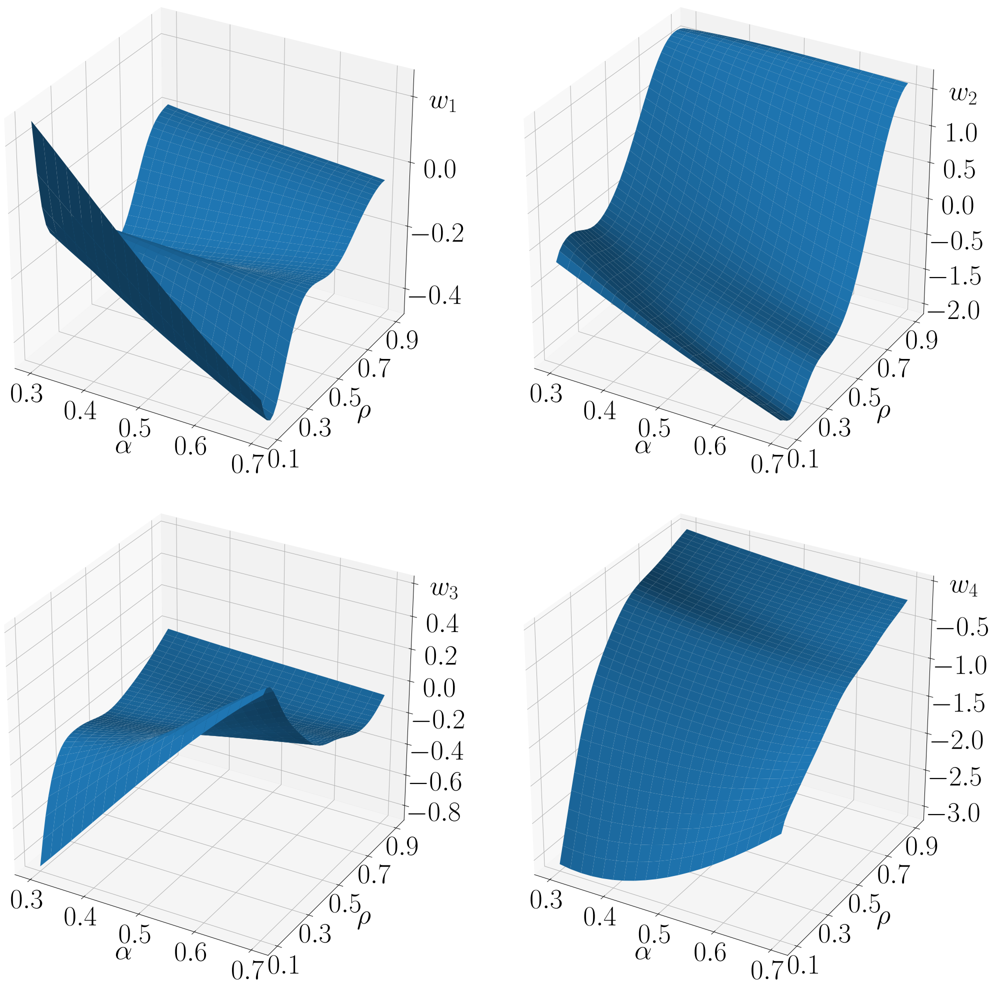


Figure 4.12: The changes in four randomly chosen elements of the $\mathbf{W}_{out}^{\alpha} = \mathbf{W}_{out}^{\mathcal{A}_1,\mathcal{A}_2}$ matrix with respect to changes in α and ρ.

be seen as providing a smooth map from the hyperparameters ρ and α to a matrix $\mathbf{W}_{out}^{\alpha}$ in the $\mathbb{R}^{D\times 2N}$ space. Therefore by considering these hyperparameters, ρ and α, this mapping generates a two-dimensional structure in the $\mathbb{R}^{D\times 2N}$ space.

With ρ and α now identified as bifurcation parameters of the RC, the focus moves towards tracking the evolution of these UAs and identifying some of these 'behind-the-scenes' bifurcations.

4.5 Bifurcations of Untrained Attractors

In this section the behaviour of the UAs found in the previous section are examined in the $(\alpha,\, x_3)$-plane for a given ρ and then in the (α, ρ)-plane. It is important to stress here again that in each of the numerical experiments conducted in this Chapter the same randomly initialised $\mathbf{M}$ and $\mathbf{W}_{in}$ matrices are used and $\mathbf{M}$ is rescaled accordingly to a specific ρ.

The behaviour of these UAs are tracked by initialising the state of the RC, trained for a certain α and ρ, with one of the corresponding fixed points shown in Fig. 4.11. The evolution of this fixed point with respect to α is tracked by repeating the process of incrementally changing α, retraining and initialising the state of the RC with the fixed point corresponding to the previous α. If there is a change in the dynamics of the RC, for instance, a bifurcation from fixed point to limit cycle or a period-doubling bifurcation, then the dynamics of the resulting stable attractor is tracked.

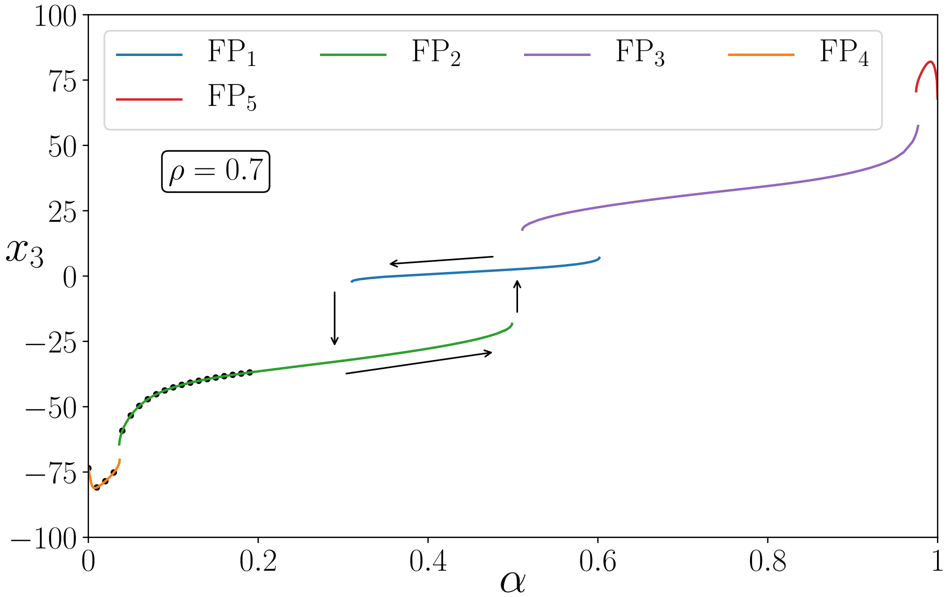
The above method of tracking the changes in the dynamics of the multifunctional RC's stable solutions is what is known colloquially amongst practitioners of nonlinear dynamics as 'poor man's' continuation due to its limitations. More advanced techniques are used in software like AUTO [134] that can track unstable solutions and capture further dynamical phenomena. However these tools are not entirely straightforward to apply to RCs as the training of the RC needs to be incorporated into the continuation procedure. While this can be done, the continuation method used here provides some insight into the dynamics of stable solutions of the multifunctional RC and adapting AUTO to incorporate RC type systems is left for future work.

4.5.1 Bifurcations in the $(\alpha,\, x_3)$-plane

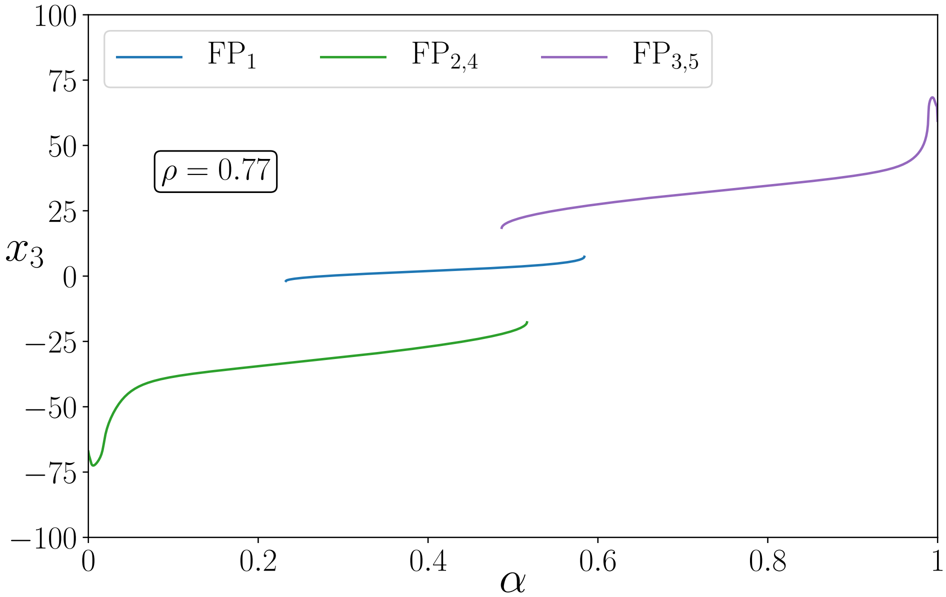
$\rho = 0.7$

The result of applying the above tracking procedure to the UAs from Fig. 4.11 for $\rho = 0.7$ is shown in Fig. 4.13a where the extent of the bistabilities in Fig. 4.11 can now be seen. Additional branches of fixed points and bistabilities closer to the end points of α are also uncovered. The evolution with respect to α of the previously found fixed point located in between the reconstructed chaotic attractors in Fig. 4.11 is now labelled as the branch FP_1 and the evolution of the fixed points closer to $\hat{\mathcal{A}}_2$ and $\hat{\mathcal{A}}_1$ are labelled as FP_2 and FP_3 respectively. The evolution of the fixed points closest to $\alpha = 0$ and 1 are denoted as the branch FP_4 and FP_5 respectively. What generates these bistabilities are the hysteresis cycles created here. When moving along the FP_2 branch as α is increased, indicated by the arrows in Fig. 4.13a, there is a point where the state of the RC jumps to the FP_1 branch. After this transition, if α were instead decreased the state of the RC then remains on the FP_1 branch to the point where the state of the RC returns to the FP_2 branch, and so on.

Fig. 4.13a also demonstrates that when the RC fails to reconstruct $\mathcal{A}_1$ then the predicted trajectory falls onto the FP_4 and FP_2 branches. The black points plotted in Fig. 4.13a are the fixed points that the predictions of $\mathcal{A}_1$ decay towards in Fig. 4.10. As these black points line up directly with the branches of FP_4 and FP_2, it can be concluded that as attractor reconstruction of $\mathcal{A}_1$ is achieved, these fixed points do not suddenly disappear and their existence is intrinsic to the dynamics of this RC setup. In modes of failure these branches of fixed points provide routes to stability which also gives greater insight to the behaviour of the RC at the boundaries of multifunctionality.



(a) $\rho = 0.7$



(b) $\rho = 0.77$

Figure 4.13: Tracking the dynamics of untrained attractors in (α, x_3)-plane for; (a) $\rho = 0.7$ and (b) $\rho = 0.77$. In (a) black arrows indicate direction of hysteresis cycle and black points correspond with the location of the fixed points in Fig. 4.10.

Evidence of bifurcations

The stable branches of equilibria depicted in Fig. 4.13 would ordinarily be connected by branches of unstable equilibria but given the nature of the tracking approach used here, these cannot be explicitly determined. However, as the end points of these branches are seemingly being drawn closer together, i.e., the right and left end points of both FP_4 and FP_2 and likewise for FP_3 and FP_5, this can be inferred as a signature of the existence of unstable branches of equilibria and evidence that at these end points are saddle-node (SN) bifurcations.

$\boldsymbol{\rho = 0.77}$

To help strengthen the above claim, the same tracking procedure is applied to the RC when $\rho = 0.77$. The result of this is shown in Fig. 4.13b where the right and left end points of FP_4 and FP_2 as well as FP_3 and FP_5 have become connected thereby resulting in two branches of fixed points labelled as $\mathrm{FP}_{2,4}$ and $\mathrm{FP}_{3,5}$. As the branches connect this particular behaviour is indicative of two cusp bifurcations taking place about $\alpha \approx 0.034$ and 0.988 as ρ is increased from 0.7 to 0.77. In addition, a region of 'tristability' is shown here for $0.4867 \leq \alpha \leq 0.5165$ where there is an overlap between all three branches.

4.5.2 Bifurcations in the (α, ρ)-plane

Given the insight that these behind-the-scenes bifurcations have provided so far, the same method is used to explore and characterise the behaviour of these UAs by tracking their evolution in the (α, ρ)-plane. The result of this is depicted in Fig. 4.14.

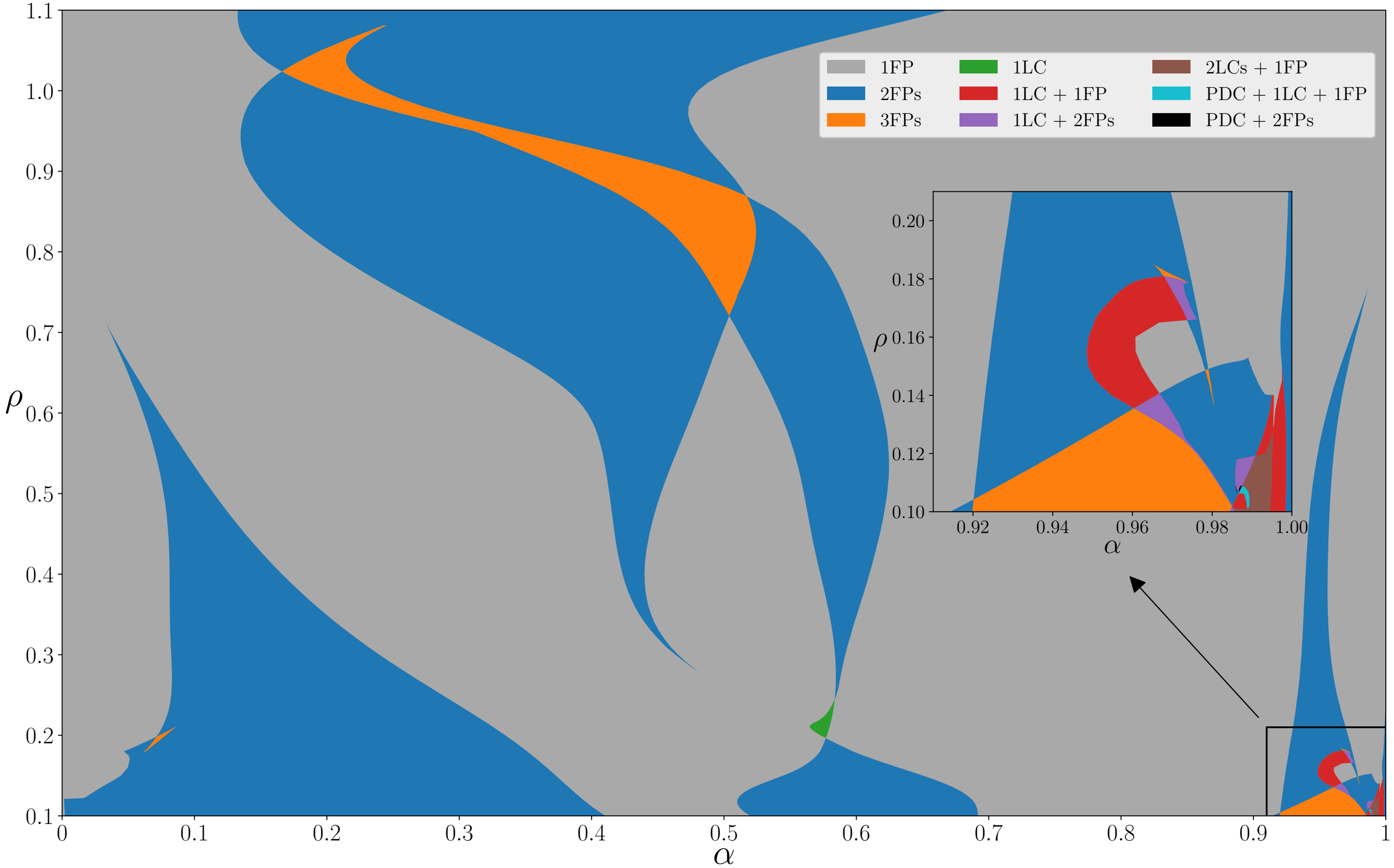


Figure 4.14: Classification of untrained attractors in the $(\alpha,\, \rho)$-plane. (FP: Fixed Point, LC: Limit Cycle, PDC: Period Doubling Cascade)

Cusp bifurcations

Here the previously mentioned cusp bifurcations can be seen taking place at $(\alpha, \rho) = (0.0337, 0.71)$ and $(0.9875, 0.76)$. Another cusp bifurcation of the FP_1 and FP_2 branches takes place at $(\alpha, \rho) \approx (0.48, 0.27)$. $FP_{1,2}$ is defined as the branch of stable fixed points emerging from this cusp bifurcation.

Limit cycles ($\rho = 0.21$)

Fig. 4.14 also reveals the existence of several limit cycles in $\mathbb{P}$. As ρ is decreased from 0.27, the bistability between FP_3 and $FP_{1,2}$ is lost at $(\alpha, \rho) \approx (0.5837, 0.2449)$. Fig. 4.15 also shows that as these branches begin to drift apart for $\rho = 0.21$ a period-1 limit cycle, LC_1, is born in this gap. Plotted here is the evolution of the maximum and minimum values of LC_1 versus α. As ρ is decreased further, the bistability between the FP_3 and $FP_{1,2}$ branches resumes from $(\alpha, \rho) \approx (0.577, 0.1965)$ resulting in the death of LC_1.

Tristability

The extent of the region of tristability found in Fig. 4.13b is shown in Fig. 4.14.

Another relatively small region of tristability between $FP_{1,2}$, FP_7 and FP_4 is found in Fig. 4.15. While the left end of $FP_{1,2}$ had at one time appeared to be growing towards and potentially connecting to FP_4 in another cusp bifurcation, instead it has broken off due to a cusp bifurcation taking place on $FP_{1,2}$ at $(\alpha, \rho) \approx (0.087, 0.212)$, as shown in Fig. 4.14. This results in a new branch of fixed points FP_7 as seen in Fig. 4.15. The existence of FP_7 is relatively brief as its own end points are quickly drawn together and disappear entirely at $(\alpha, \rho) \approx (0.062, 0.179)$ in Fig. 4.14. This particular behaviour also occurs on FP_3 where a more detailed picture and explanation of these events is provided later in this section. The beginnings of a branch of stable fixed points can be seen emerging from the right hand side of Fig. 4.15, these are denoted as FP_6, some interesting properties emanating from this branch are also shown later in the present section.

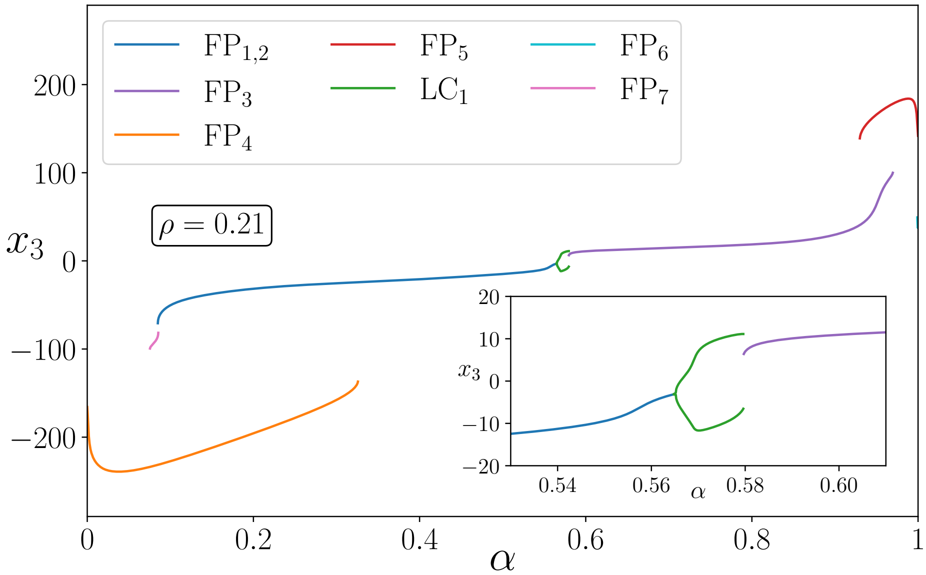


Figure 4.15: Tracking untrained attractors for $\rho = 0.21$: Max and min values of limit cycle, LC_1, born at the right and left end points of the $FP_{1,2}$ and FP_3 branches. Also seen are the FP_4, FP_5, FP_6 and FP_7, branches of stable fixed points.

Bifurcations for large α and small ρ

Given the abundant variety of behaviour exhibited by the UAs for large α and small ρ, as seen in the inset plot of Fig. 4.14, a more expansive picture is provided in Fig. 4.16 which depicts the events leading towards certain bifurcations. The cusp bifurcation taking place at $(\alpha, \rho) = (0.965, 0.185)$ gives rise to a relatively small tristable region. This cusp bifurcation originates from a buckling of the FP_3 branch where a new branch of stable fixed points emerges, labelled as FP_8 in Fig. 4.16. A limit cycle is born at the right end point of FP_3 for $\rho \approx 0.181$. Fig. 4.16 shows the evolution of the maximum and minimum x_3-values of this period-1 limit cycle, labelled as LC_2. Close to this cusp bifurcation it can be seen that for $\rho = 0.18$, LC_2 is contained within a Hopf-bubble as it exists between two supercritical Hopf bifurcations. However, as ρ is decreased further, the amplitude of oscillation of LC_2 increases and the bubble bursts. It is also found that LC_2 undergoes period-doubling (PD) bifurcations, the first of which occurs nearby $(\alpha, \rho) = (0.959, 0.166)$. These PD

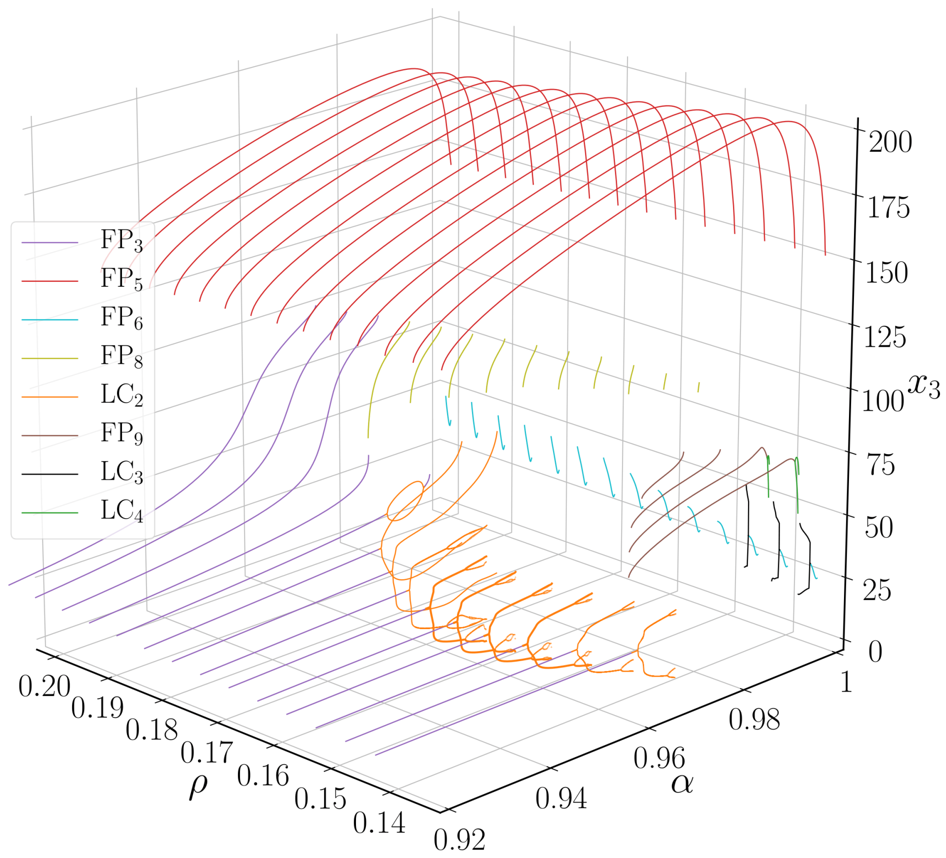


Figure 4.16: Tracking of the specified untrained attractors at large α for different ρ values.

bifurcations contribute to the significant reduction in α-values for which LC_2 can be tracked. Also shown in Fig. 4.16 are cases where LC_2 subsequently encounters another PD bifurcation resulting in a period-4 limit cycle.

The existence of FP_8 throughout Fig. 4.16 is relatively brief, as ρ is decreased its end points are drawn closer together until this branch becomes but a single point-like attractor at $(\alpha, \rho) \approx (0.98, 0.137)$ and can no longer be tracked. This event and another mentioned earlier regarding FP_7 is evidence that further flavours of cusp-like bifurcations take place in $\mathbb{P}$. A codimension-3 bifurcation analysis (involving either σ, β or γ) could, for example, unveil a swallowtail bifurcation point (the point at which two cusp branches collide). For reading on more cusp-like bifurcations see [135] or [136]. Alternatively, if swallowtail bifurcation points were found to take place in lower dimensional representations of Eq. (2.19) (for $N = 2, 3, \ldots$), then their existence could be generalised to the higher dimensional picture. The benefit of reducing the dimension for such an analysis would allow for the use of numerical continuation software like AUTO [137] on Eq. (2.19). Moreover, this approach would facilitate the study of unstable attractors and identify further dynamical features inherent to Eq. (2.19).

As shown in Fig. 4.16, another branch of fixed points, labelled as FP_9, is found. For $\rho = 0.135$, at the right end point of FP_9 a limit cycle labelled as LC_4 is born. Increasing ρ results in the loss of LC_4 as the end points of FP_9 are being drawn closer together where eventually at $(\alpha, \rho) = (0.989, 0.153)$ it can no longer be tracked.

Untrained chaotic attractor

Shown throughout Fig. 4.16 is the evolution of FP_6, the branch previously mentioned regarding the discussion of Fig. 4.15. As ρ is decreased, a period-1 limit cycle, labelled as LC_3, is born from the left end point of FP_6. The brown region in Fig. 4.14 depicts the coexistence of LC_3, LC_4, and FP_5. However, as the evolution of the local maxima and minima of LC_3 are tracked there are certain points in the $(\alpha,\, \rho)$-plane where LC_3 undergoes a PD bifurcation. Furthermore, there at times at which one PD bifurcation leads to another and in turn triggers a period-doubling cascade (PDC) ultimately resulting in chaotic behaviour.

Two relatively small regions in Fig. 4.14 (coloured in black and cyan) highlight where PD bifurcations of LC_3 lead to PDCs. An example of this particular sequence of PD bifurcations is shown in Fig. 4.17 when setting $\rho = 0.107$ and highlights how FP_6 evolves as α is decreased from 1. Plotted here are the evolution of the local maxima and minima of LC_3 which sprouts from the left end point of FP_6 for $\alpha = 0.9986$. The dynamics of LC_3 are tracked as α is reduced further. The first PD bifurcation occurs at $\alpha \approx 0.989$ where there are now two distinct local maxima and minima of LC_3. This particular behaviour is depicted below the bifurcation diagram in Fig. 4.17 with snapshots of LC_3 as it travels along its route to a PDC in the (x_1, x_3)-space of $\mathbb{P}$. Here, it can be seen how LC_3 transitions from period-1 at $\alpha = 0.9895$ to period-2 at $\alpha = 0.9885$. Decreasing α further to 0.9875 an infinite number of PD bifurcations has taken place resulting in the birth of an untrained chaotic attractor. Periodic behaviour briefly resumes for decreasing α further before entering another PDC as a second bout of chaos begins at $\alpha = 0.98618$ and ends at 0.98638.

Despite that multifunctionality is not achieved in the relatively small regions in which the PDCs are shown in Fig. 4.14, it remains a remarkable result which, similar to Fig. 2.6, shows that when attempting to train the RC to provide a coexistence of two desired chaotic attractors it is also possible for another chaotic attractor to exist in the background. Moreover, for a different choice of the other RC parameters, these PDCs could play a larger role in $\mathbb{P}$ and further influence the RC's ability to reconstruct a given attractor.

General Comments

Overall, the long term characterisation of the RC's prediction of a given attractor in Figs. 4.6a-4.6b combined with the classification of the UAs in Fig. 4.14 provides a clear picture of $\mathbb{P}$ for a given α and ρ. When put together these describe the regions in which multifunctionality was achieved and the type of untrained attractor that the RC can approach when attractor

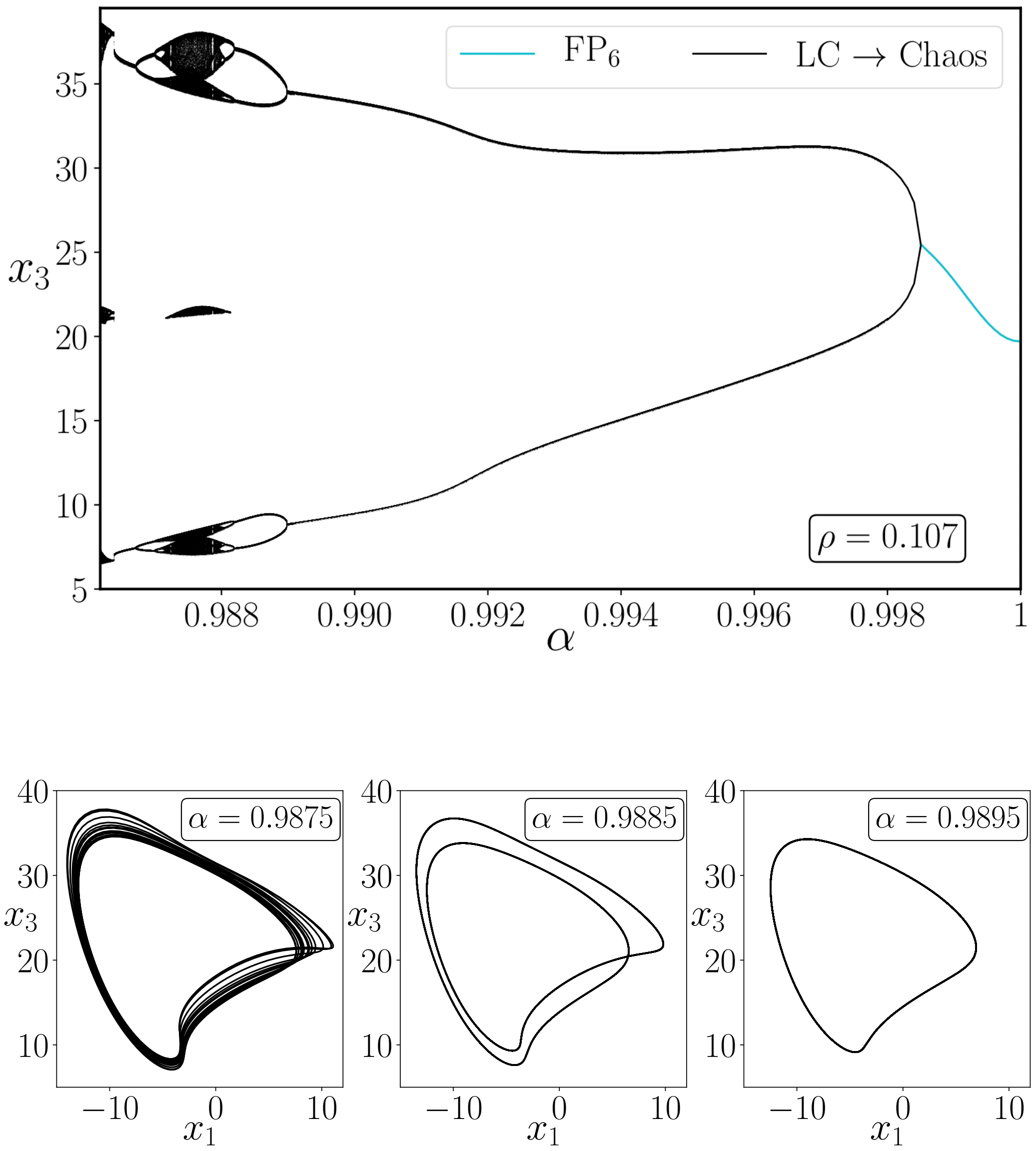


Figure 4.17: ($\rho = 0.107$): Evolution of FP_6 as it bifurcates to a limit cycle and then to a chaotic attractor through an infinite sequence of period-doubling bifurcations. Also shown are snapshots of this limit cycle along its route to chaos.

reconstruction fails.

As a final comment, throughout the results in this Chapter, the random topology of the RC matrices, $\mathbf{M}$, and $\mathbf{W}_{in}$ have remained fixed after an initial realisation and the same matrices were used to generate the results regarding Case I-III. Given a different initialisation of these matrices it is possible that quantitative changes in these bifurcation figures occur. From further analysis and results which are not shown here, it is found that the main qualitative features of the RC's dynamics continue to survive.

4.6 Conclusions

The results shown in this Chapter demonstrate that a RC can be trained to exhibit multifunctionality whereby, for a given IC, the dynamics of more than one chaotic attractor can be successfully reconstructed in $\mathbb{P}$.

In order to train a RC to express multifunctional dynamics, the 'blending technique' introduced in Sec. 3.2.1 is used as a means to combine and weight data from two different sources. The flexibility of this technique is tested by training the RC to reconstruct a coexistence of chaotic attractors from: a system which already exhibits multistability, two parameter settings of a system and, two different systems entirely.

The results in this Chapter illustrate that the coexistence of attractors which the multifunctional RC reconstructs are not limited to the one system. In Fig. 4.5b it is shown that the RC was successfully trained to permit the coexistence of the Lorenz butterfly attractor, $\mathcal{L}$, and the chaotic attractor $\mathcal{A}_2$ generated from Eq. (4.1). From this comes the question on the amount of attractors that can be successfully trained to coexist in $\mathbb{P}$ and steps are taken to address this in later Chapters.

The ability to combine attractors from different sources to then coexist in the same state space broadens the current set of RC applications. This demonstrates that instead of having to change parameters in a system for it to exhibit a different behaviour, the various desired modes of operation can coexist in the state space of a multifunctional RC where an appropriate controller could in theory be designed to switch between attractors [133]. As discussed in Sec. 3.4.2, such a control mechanism draws further parallels to the role of neuromodulators in biological neural networks.

In order to achieve the desired training outcome, there is a crucial dependence on certain RC design and training parameters, in particular the 'blending' parameter, α, and, ρ, the spectral radius of the RC's internal connection matrix. When varying these parameters, an abundance of nontrivial transitions between multifunctionality and modes of failure occur as there is a competition between attractors in $\mathbb{P}$ for a given choice of α and ρ. For example, when attempting to reconstruct a coexistence of attractors there are times when the RC is able to reconstruct the dynamics of only one chaotic attractor or in further events of failure, the predicted trajectory on a given chaotic attractor can tend toward a limit cycle or fixed point.

On closer inspection, it is found that even when the RC is successfully trained to achieve multifunctionality there are also many UAs in $\mathbb{P}$. These attractors and their basins of attraction inhabit large regions of $\mathbb{P}$ but were not part of the training and limit the regions in which multifunctionality is achievable. Moreover, these UAs are shown to have dynamics of their own and play a significant role in the event that the desired attractor cannot be successfully reconstructed.

By tracking the evolution of these UAs with respect to α and ρ, a number of 'behind-the-scenes' bifurcations have been identified. In particular when tracking the evolution of FP_6 it is shown in Fig. 4.17 how this branch of stable fixed points transitions to the limit cycle LC_3 which subsequently undergoes a series of period-doubling bifurcations to the point of provoking a period-doubling cascade leading towards the creation of a chaotic attractor. So while the objective is to train the RC to reconstruct a coexistence of two specific chaotic attractors in $\mathbb{P}$ there is another chaotic attractor created in a *sub rosa* fashion.

Much like the investigation of the UAs in $\mathbb{P}$, there is also evidence of similarly occurring events in the brain which influence and disrupt its normal behaviour. It is understood that neurological disorders like Parkinson's disease or epilepsy occur as a result of certain active regions of the brain deviating from its normal state of operation and then becoming trapped within some undesirable behaviour beyond which it may not be able to resume its normal function. The effort is often made to model and control these events in order to counteract the effects of these illnesses [138–140].

Overall, the results in this Chapter emanate from employing the dyadic ‘two-way street’ approach of framing neurological features in the context of dynamical systems. Moreover, this work indicates that if other hypotheses or known facets of the brain can be articulated in this manner then there lies the potential to portray it artificially.

Chapter 5

Symmetry and Multifunctionality

Symmetry is a notorious issue in the training of a RC. For example, when training RCs with symmetry in their designs to mimic trajectories on a given attractor, a 'mirrored' version of the attractor is automatically created in the RC's state space. In this Chapter, the opposite scenario is considered by including certain symmetries in the training data which as a result exposes further fundamental properties of a multifunctional RC.

5.1 Issues related to Symmetry

5.1.1 Mirror-Attractors

An important property to consider when designing a RC is whether there is a symmetry present in its formulation. Symmetry introduces certain problems which can stifle the RC's learning capability and dramatically reduce its reconstruction capacity. For example, if $\boldsymbol{q}(\hat{\boldsymbol{r}}(t)) = \hat{\boldsymbol{r}}(t)$ instead of setting $\boldsymbol{q}(\hat{\boldsymbol{r}}(t))$ as in Eq. (2.13), a symmetry in Eq. (2.19) of the following form arises,

$$\dot{\hat{\boldsymbol{r}}}(t) = \gamma \left[-\hat{\boldsymbol{r}}(t) + \tanh\left(\mathbf{M}\, \hat{\boldsymbol{r}}(t) + \sigma \mathbf{W}_{in} \mathbf{W}_{out}\, \hat{\boldsymbol{r}}(t) \right) \right], \tag{5.1}$$

which is invariant under inversion of the sign $\hat{\boldsymbol{r}}(t) \rightarrow -\hat{\boldsymbol{r}}(t)$. This results in a symmetric partner for every solution of Eq. 5.1, whereby for any attractor, $\mathcal{S}$, in $\mathbb{S}$ there is a corresponding 'mirror-attractor', $\mathcal{S}'$. Additionally, the linear readout results in pairs of reconstructed attractors and corresponding mirror-attractors ($\hat{\mathcal{A}}$ and $\hat{\mathcal{A}}'$) appearing in $\mathbb{P}$. Herteux and Räth were the first to explore the issues related to mirror-attractors in [77].

Mirror-attractors become most problematic when the location of the training data is not chosen correctly as the attractors in the $\mathbb{S}$ may merge with one another. For instance, if a given $\hat{\mathcal{A}}$ and the corresponding $\hat{\mathcal{A}}'$ share mutual regions of $\mathbb{P}$, i.e., $\hat{\mathcal{A}} \cap \hat{\mathcal{A}}' \neq \emptyset$, and the RC has not been trained to support this form of multistable dynamics, the resulting dynamics of the RC could resemble the results shown in Fig. 2 in [77].

Breaking the Symmetry

This symmetry induced phenomenon and in particular how to overcome the issues related to it was explored in great detail in [77]. Here it was proven analytically, under the assumption that the RC has the echo state property (see Sec. 2.2.2 for further details), that by changing the sign of the input and target data in the listening stage, this leads to a sign change in the resulting RC state variables used as training data. As a consequence, the resulting readout matrix $\mathbf{W}_{out}$ will have the same weights independent of the sign change.

There are numerous ways to break this symmetry and mitigate the restraints it imposes. For example, using a bias in the output, a shift in the input, or including square terms in the readout. The focus of this Chapter is on the square terms in the readout described by Eq. (2.13). In this case the RC is not invariant under the change of sign like before as,

$$\begin{aligned}\hat{\psi}\left(-\hat{\boldsymbol{r}}(t)\right) &\approx \mathbf{W}_{out}^{(1)}\left(-\hat{\boldsymbol{r}}(t)\right) + \mathbf{W}_{out}^{(2)}\left(-\hat{\boldsymbol{r}}(t)\right)^2 \\ &= -\mathbf{W}_{out}^{(1)}\hat{\boldsymbol{r}}(t) + \mathbf{W}_{out}^{(2)}\hat{\boldsymbol{r}}^2(t) \neq -\hat{\psi}\left(\hat{\boldsymbol{r}}(t)\right),\end{aligned} \tag{5.2}$$

and $\mathbf{W}_{out}^{(2)}\hat{\boldsymbol{r}}^2(t)$ breaks the symmetry which destroys the mirror-attractors.

Example with $\mathcal{L}$

Using the same RC with $\rho = 0.15$ from Sec. 2.4 as an example, the difference between setting $\boldsymbol{q}(\boldsymbol{r}(t))$ as defined in Eq. (2.13) and $\boldsymbol{q}(\boldsymbol{r}(t)) = \boldsymbol{r}(t)$ is shown in Fig. 5.1. When the symmetry is broken, as in Eq. (2.19), the trained RC reconstructs $\mathcal{L}$ when initialised in the usual sense with $\hat{\boldsymbol{r}}(0) = \boldsymbol{r}(t_{\text{train}})$ and when the RC is initialised with $-\hat{\boldsymbol{r}}(0)$ then its state tends to the UA, LC_2, like

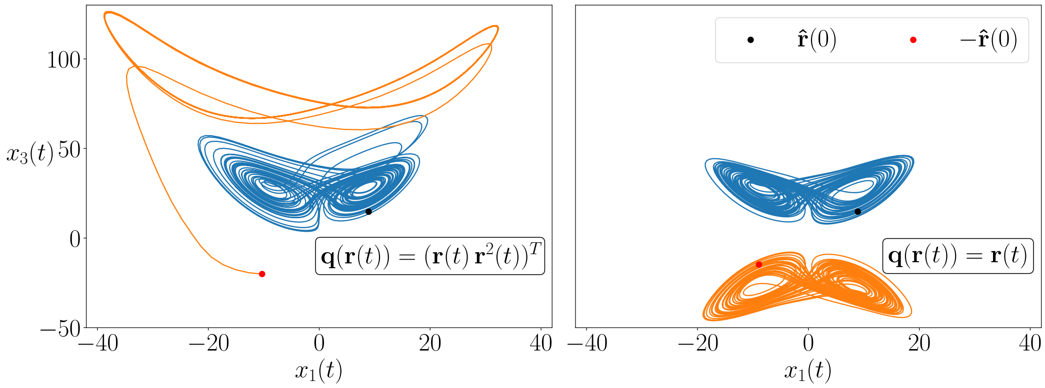


Figure 5.1: Illustrating how different choices of $\boldsymbol{q}(\boldsymbol{r}(t))$ effects the RC's dynamics when initialised from $\hat{\boldsymbol{r}}(0) = r_{\text{train}}$ and $-r_{\text{train}}$.

in Fig. 2.5. However when $\boldsymbol{q}(\boldsymbol{r}(t)) = \boldsymbol{r}(t)$, then the mirror-attractor appears when the RC is initialised with $-\hat{\boldsymbol{r}}(0)$.

5.1.2 Symmetry and Multifunctionality

Based on this information regarding mirror-attractors discussed so far, the following question naturally arises, is the opposite also true? More specifically, what happens when a RC which does not have symmetry in its design is trained on data related by symmetry?

In this Chapter, these symmetry associated phenomena are considered from the perspective of multifunctionality by specifically training the RC to reconstruct a coexistence of attractor and mirror-attractor pairs. In Sec. 5.2, it is proven analytically that, under certain conditions, symmetries in the training data prohibit $\mathbf{W}_{out}^{(2)}$ from existing and the consequences of this analytical result are explored in various numerical experiments.

5.1.3 Previous RC Symmetry studies

Before discussing the main results of the present Chapter, it is important to highlight that the effect of symmetries in RC has been investigated in a number of previous studies. Carroll and Pecora show in [86] how symmetries in the structure of the network lead to a lower covariance rank and negatively affect the training error. A permutation symmetry in the RC states was found to severely impact the performance of a swarm based RC as studied in [141]. On the other hand, matching the symmetry of the input data with a suitable handcrafted RC design was shown to cause drastic improvements in two tasks in [142]. In contrast, the focus of this Chapter is to exam how a RC can adapt to and take on the symmetry of the trained system spontaneously.

5.2 Analytical Results

In this section, it is proven analytically that, under certain conditions, a point or inversion symmetry of the training data forbids $\mathbf{W}_{out}^{(2)}$ to exist.

5.2.1 Anti-symmetric training data leads to $W_{out}^{(2)} = 0$

Let's now study the properties of the ridge regression formula in Eq. (2.16) if it is presented with anti-symmetric data. Assuming that $\mathbf{X}$ and $\mathbf{Y}$ both have $2n$ columns and are obtained from $\boldsymbol{r}(t_i)$ and $\boldsymbol{u}(t_i)$ data with the property,

$$\boldsymbol{u}(t_{i+n}) = -\boldsymbol{u}(t_i), \tag{5.3}$$

$$\boldsymbol{r}(t_{i+n}) = -\boldsymbol{r}(t_i). \tag{5.4}$$

Then for $i = 1, \dots, n$,

$$\mathbf{X} = \left[\begin{pmatrix} \boldsymbol{r}(t_1) \\ \boldsymbol{r}^2(t_1) \end{pmatrix} \cdots \begin{pmatrix} \boldsymbol{r}(t_n) \\ \boldsymbol{r}^2(t_n) \end{pmatrix} \begin{pmatrix} -\boldsymbol{r}(t_1) \\ \boldsymbol{r}^2(t_1) \end{pmatrix} \cdots \begin{pmatrix} -\boldsymbol{r}(t_n) \\ \boldsymbol{r}^2(t_n) \end{pmatrix} \right], \tag{5.5}$$

$$\mathbf{Y} = \left[\begin{matrix} \boldsymbol{u}(t_1), & \cdots, & \boldsymbol{u}(t_n), -\boldsymbol{u}(t_1), & \cdots, & -\boldsymbol{u}(t_n) \end{matrix} \right]. \tag{5.6}$$

As a result, the $\mathbf{YX}^T$ and $\mathbf{XX}^T$ terms used in the calculation of $\mathbf{W}_{out}$ in Eq. (2.16) can now be written as,

$$\mathbf{YX}^T = \left[\begin{matrix} 2\sum_{t_i=t_{\text{listen}}}^{t_{\text{train}}} \boldsymbol{u}(t_i)\left(\boldsymbol{r}(t_i)\right)^T, & \mathbf{0} \end{matrix} \right], \tag{5.7}$$

$$\mathbf{XX}^T = \left[\begin{matrix} 2\sum_{t_i=t_{\text{listen}}}^{t_{\text{train}}} \boldsymbol{r}(t_i)\left(\boldsymbol{r}(t_i)\right)^T & \mathbf{0} \\ \mathbf{0} & 2\sum_{t_i=t_{\text{listen}}}^{t_{\text{train}}} \boldsymbol{r}^2(t_i)\left(\boldsymbol{r}^2(t_i)\right)^T \end{matrix} \right]. \tag{5.8}$$

Plugging Eqs. 5.7-5.8 into Eq. (2.16) results in,

$$\mathbf{W}_{out} = \left[2 \left(\sum_{t_i = t_{\text{listen}}}^{t_{\text{train}}} \boldsymbol{u}(t_i)(\boldsymbol{r}(t_i))^T \right) \left(2 \sum_{t_i = t_{\text{listen}}}^{t_{\text{train}}} \boldsymbol{r}(t_i)(\boldsymbol{r}(t_i))^T + \beta \mathbf{I} \right)^{-1}, \ \mathbf{0} \right]. \tag{5.9}$$

Therefore, whenever the training data fulfils the conditions in Eqs. (5.3) and (5.4), it follows that,

$$\mathbf{W}_{out}^{(2)} = \mathbf{0}. \tag{5.10}$$

5.2.2 Sources of anti-symmetric training data

Multiple trajectories related by symmetry

Since it has now been proven that anti-symmetric training data leads to a vanishing $\mathbf{W}_{out}^{(2)} = \mathbf{0}$, the next step is to investigate, under what conditions such anti-symmetric training data naturally arises. In the context of multifunctionality, this can happen if the RC is trained with input from two attractors $\boldsymbol{u}_1(t)$ and $\boldsymbol{u}_2(t)$ which are related by,

$$\boldsymbol{u}_1(t) = -\boldsymbol{u}_2(t). \tag{5.11}$$

Time discretisation and concatenation of the two trajectories according to the blending technique of Eq. (3.13) with $\alpha = 0.5$ then immediately shows that Eq. (5.3) holds. To show that Eq. (5.4) is also fulfilled, both $\boldsymbol{r}_1(t)$ and $\boldsymbol{r}_2(t)$ are obtained from Eq. (2.10). Let's define,

$$f(\boldsymbol{r}, \boldsymbol{u}) = \gamma \left[-\boldsymbol{r} + \tanh \left(\ \mathbf{M} \, \boldsymbol{r} + \sigma \mathbf{W}_{in} \, \boldsymbol{u} \ \right) \right] = -f(-\boldsymbol{r}, -\boldsymbol{u}), \tag{5.12}$$

where in the second equation the resulting f is anti-symmetric. If now $\boldsymbol{r}_1(t)$ is a solution of Eq. (2.10) for some IC, $\boldsymbol{r}_1(0)$, then $\boldsymbol{r}_2(t) = -\boldsymbol{r}_1(t)$ is a solution for the following IC,

$$\boldsymbol{r}_2(0) = -\boldsymbol{r}_1(0). \tag{5.13}$$

This is demonstrated by,

$$\dot{\boldsymbol{r}}_2(t) = -\dot{\boldsymbol{r}}_1(t) = -f(\boldsymbol{r}_1(t), \boldsymbol{u}_1(t)), \tag{5.14}$$

$$= -f(-\boldsymbol{r}_2(t), -\boldsymbol{u}_2(t)) = f(\boldsymbol{r}_2(t), \boldsymbol{u}_2(t)). \tag{5.15}$$

Therefore Eq. (5.4) holds. This means that if the RC is trained with two distinct anti-symmetric trajectories, and the IC of the RC state is anti-symmetric, the resulting $\mathbf{W}_{out}^{(2)}$ matrix automatically vanishes. This result can be easily generalised to any even number of distinct trajectories $\boldsymbol{u}_k(t)$, which are pairwise anti-symmetric, i.e., that are of the following form,

$$\boldsymbol{u}_{2k}(t) = -\boldsymbol{u}_{2k+1}(t). \tag{5.16}$$

Single anti-symmetric trajectory

Another source of anti-symmetric data is the case of periodic trajectories of period T which are of the form,

$$\boldsymbol{u}\left(t + \frac{T}{2}\right) = -\boldsymbol{u}(t). \tag{5.17}$$

As the RC's state, $\boldsymbol{r}(t)$, is the response of the system to the periodic drive $\boldsymbol{u}(t)$ according to Eq. (2.10), by the above symmetry there always exists a solution of the form,

$$\boldsymbol{r}\left(t + \frac{T}{2}\right) = -\boldsymbol{r}(t), \tag{5.18}$$

however, this solution is not necessarily stable. Later, in the results regarding the numerical experiments in Sec. 5.3.3, it is found that the RC's dynamics is stable up to a certain value of ρ and in this case Eq. (5.4) is fulfilled which again gives rise to vanishing $\mathbf{W}_{out}^{(2)}$.

5.3 Numerical Results

The analytical results from Sec. 5.2 are now used as a platform to extend the study of the relationship between symmetry and multifunctionality through several numerical experiments. These experiments consist of training the RC to specifically reconstruct a coexistence of the Lorenz attractor and its mirror-attractor. A paradigmatic example of anti-symmetric training data is also explored numerically.

Figure	ρ	σ	γ	β	α	t_{listen}	t_{train}	t_{predict}
5.2-5.5	0.2	0.2	10	10^{-2}	0.5	100	200	300
5.6	1.43	0.2	10	10^{-2}	0.5	100	200	300
5.7	1.7	0.3	5	10^{-1}	0.5	100	200	300
5.8	1.7	0.22	5	10^{-1}	n/a	100	200	300
5.9-5.10	$[1.4, 2.4]$	0.3	10	10^{-2}	n/a	$30T$	$60T$	$90T$

Table 5.1: Hyperparameters and time parameters used to generate the results displayed in each of the specified Figures. T is the period of the circular orbit described in Eq. (5.20). These parameters were chosen using a grid search method.

The RC hyperparameters used to produce each of the figures presented in the present section are given in Table 5.1. Also note that in each case the predicted trajectory for a given input/attractor, $\mathcal{A}$, is represented by $\hat{\mathcal{A}}$ which is in keeping with the notation used throughout this thesis.

Furthermore, the numerical results illustrated in Figs. 5.2-5.8 are generated using the same $\mathbf{M}$ and $\mathbf{W}_{in}$, with $N = 1000$, in order to provide consistency in the discussion of the following results when relating one result to another. The argument near the end of Sec. 4.5 also applies here where relatively small quantitative changes occur in the results when using different initialisations of $\mathbf{M}$ and $\mathbf{W}_{in}$ however the main characteristics of the results remain similar. The same is said for Figs. 5.9-5.10 however the dimension of $\mathbf{W}_{in}$ is decreased in order to take into consideration the lower dimensional training data.

5.3.1 Multifunctionality with mirrored attractors

The case described in Eq. (5.11) is examined numerically by training the RC in Eq. (2.10) with the squaring technique in Eq. (2.13) to reconstruct the chaotic Lorenz attractor, $\mathcal{L}$, and its mirror-attractor, $\mathcal{L}'$. The input data is generated using the 4^{th} order Runge-Kutta method with time-step, $\tau = 0.01$. The mirror-attractor is created by taking the negative of the Lorenz input time-series, i.e., $\boldsymbol{u}_{\mathcal{L}'}(t) = -\boldsymbol{u}_{\mathcal{L}}(t) = (x_1(t),\, x_2(t),\, x_3(t))^T$ where $x_1(t)$, $x_2(t)$, $x_3(t)$ are the dynamical variables of the Lorenz system described in Eq. (2.28).

The methods outlined in Sec. 3.2.1 to calculate $\mathbf{W}_{out} = \left(\mathbf{W}_{out}^{(1)}\,\mathbf{W}_{out}^{(2)}\right)$ are applied here again in order to train the RC to exhibit multifunctionality by reconstructing either $\mathcal{L}$ or $\mathcal{L}'$ based on a given IC.

Reconstructing a coexistence of $\mathcal{L}$ and $\mathcal{L}'$

In Fig. 5.2a the reconstructed attractors, $\hat{\mathcal{L}}$ and $\hat{\mathcal{L}}'$, are shown to coexist in the prediction state space, $\mathbb{P}$. The target attractors $\mathcal{L}$ and $\mathcal{L}'$ are also plotted to serve as a comparison. Fig. 5.2b shows that despite breaking the symmetry in the training by using the squaring technique defined in Eq. (2.13), the square readout matrix, $\mathbf{W}_{out}^{(2)}$, does not come into existence. This in turn verifies the analytical results in Sec. 5.2, the symmetry in the training data forces the RC to also become symmetric.

Breaking Symmetry: Shifting Training Data

The effects of breaking the symmetry in the training data are now considered by shifting $\mathcal{L}'$ by a factor χ in the x_1-direction, the input data in this case is written as $\boldsymbol{u}_{\mathcal{L}'_\chi}(t) = (-x_1(t) - \chi, -x_2(t), -x_3(t))^T$. The result of this experiment is displayed in Figs. 5.2c-5.2d for $\chi = 2$. Here it can be seen that the RC exhibits multifunctionality as it can reconstruct both $\mathcal{L}$ and $\mathcal{L}'_\chi$ using the same readout matrix. However, Fig. 5.2d illustrates that once the symmetry in the training data is broken by slightly shifting the location of the mirror-attractor in state space then $\mathbf{W}_{out}^{(2)}$ comes into existence.

$\mathbf{W}_{out}^{(2)}$ elements vs. χ

The results from a more extensive study concerning the behaviour of $\mathbf{W}_{out}^{(2)}$ when shifting the location of the mirror-attractor for $\chi \in [-5, 5]$ are now outlined. In Fig. 5.3 histograms of $\mathbf{W}_{out}^{(2)}$ as a function of χ are presented. Here, each colour represents the number of $\mathbf{W}_{out}^{(2)}$ elements at a specific value, the more elements there are at a certain value the darker the colour is. For $\chi = 0$ the known result from Fig. 5.2b occurs where all elements of $\mathbf{W}_{out}^{(2)}$ equal to 0. This figure also indicates that elements of $\mathbf{W}_{out}^{(2)}$ grow larger as

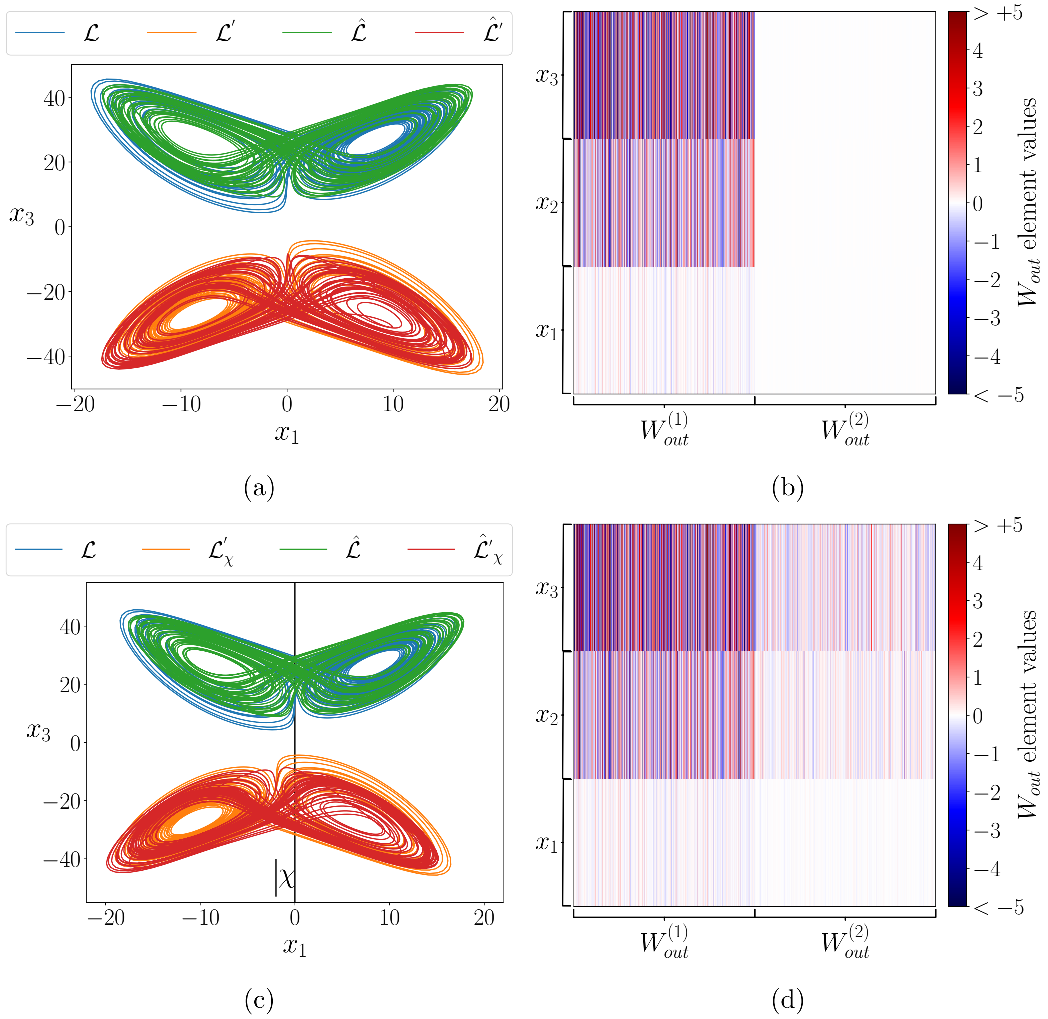


Figure 5.2: Illustrating the behaviour of $\mathbf{W}_{out}^{(2)}$ when the RC is trained to exhibit multifunctionality in cases of mirror-attractors. (a) Reconstruction of $\mathcal{L}$ and its mirror-attractor $\mathcal{L}'$ and in (b) colour plot of the corresponding trained $\mathbf{W}_{out}$ matrix individual elements. Each of the i rows in $\mathbf{W}_{out}$ is denoted by the variable it reconstructs. Each column represents the weight, $w_{i,j}$, given to the j^{th} component of $\boldsymbol{q}(\boldsymbol{r}(t)) = (\boldsymbol{r}(t), \boldsymbol{r}^2(t))^T$. The distinction is made between which columns belong to $\mathbf{W}_{out}^{(1)}$ and $\mathbf{W}_{out}^{(2)}$. (c) Reconstruction of $\mathcal{L}$ and the slightly shifted $\mathcal{L}'_\chi$ and in (d) the corresponding trained $\mathbf{W}_{out}$ matrix. Note, the matrix plots may vary depending on the PDF viewer used, it is recommended to use 'Adobe Reader' or 'Chrome PDF Viewer' to see these images as intended.

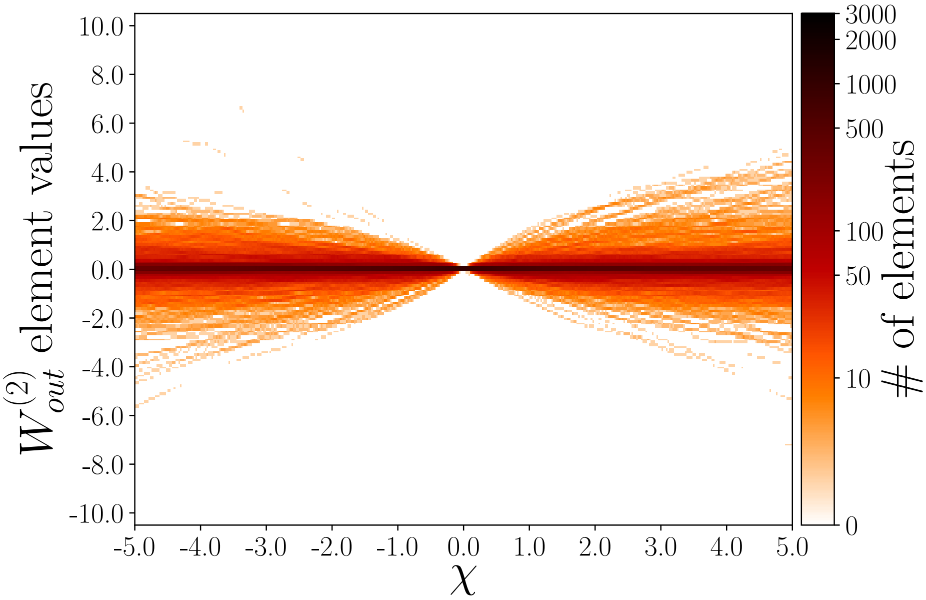


Figure 5.3: Histograms of $\mathbf{W}_{out}^{(2)}$ as a function of χ.

the magnitude of χ is increased. So the more the symmetry is broken in the training data, the greater the contribution from $\mathbf{W}_{out}^{(2)}$ in the dynamics of the multifunctional RC.

Breaking Symmetry: Imbalance of Training Data

The conditions defined in Eqs. (5.3)-(5.4) rely on the assumption that both training data sets have precisely the same number of points. The behaviour of $\mathbf{W}_{out}^{(2)}$ as the RC is trained with input from trajectories on $\mathcal{L}$ and $\mathcal{L}'$ using the respective training times $t_{\text{train}}^{\mathcal{L}}$ and $t_{\text{train}}^{\mathcal{L}'}$ are now examined. The resultant imbalance in the data sets is characterised by the new parameter $\kappa \in [0,1]$ defined as,

$$\kappa = \frac{t_{\text{train}}^{\mathcal{L}'}}{t_{\text{train}}^{\mathcal{L}} + t_{\text{train}}^{\mathcal{L}'}}. \tag{5.19}$$

Therefore, when $\kappa = 0.5$, the conditions in Eqs. (5.3)-(5.4) are satisfied as $t_{\text{train}}^{\mathcal{L}} = t_{\text{train}}^{\mathcal{L}'}$, which results in $\mathbf{W}_{out}^{(2)} = \mathbf{0}$. Fig. 5.4 illustrates the behaviour of $\mathbf{W}_{out}^{(2)}$ for $\kappa \in [0,1]$ and $t_{\text{train}}^{\mathcal{L}} + t_{\text{train}}^{\mathcal{L}'} = 200$. This shows that for a small change

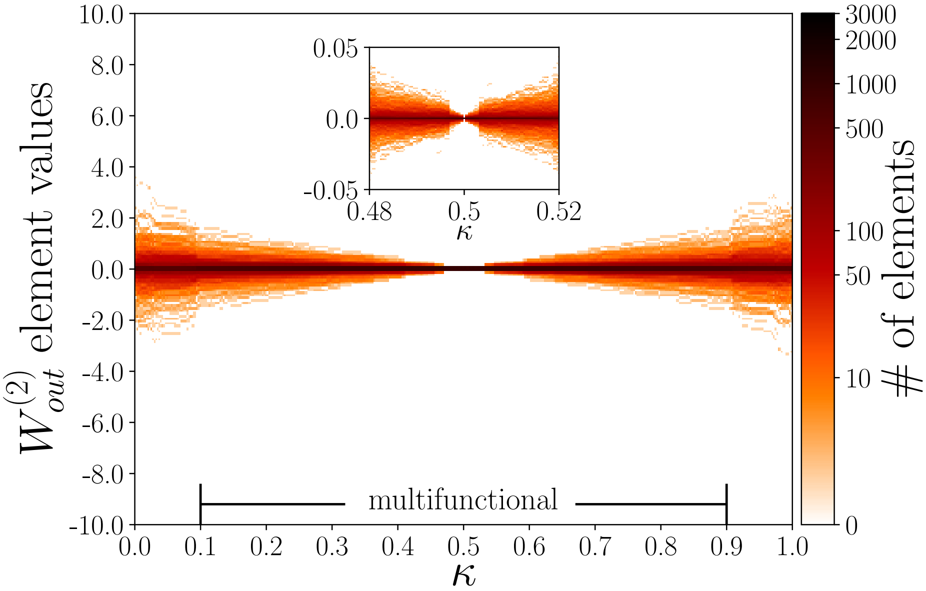


Figure 5.4: Histograms of $\mathbf{W}^{(2)}_{out}$ as a function of κ.

in κ away from 0.5 the histogram of $\mathbf{W}^{(2)}_{out}$ elements is still strongly peaked at zero.

Fig. 5.4 also highlights the range of κ values ($0.1 \leq \kappa \leq 0.9$) in which the RC is able to achieve multifunctionality. This indicates that even when there is a strong imbalance in the training data used from each attractor, the RC is still able to exhibit multifunctionality.

To provide further detail on the changes of $\mathbf{W}^{(2)}_{out}$ away from $\kappa = 0.5$, the cumulative density function of the absolute values of the elements of $\mathbf{W}^{(2)}_{out}$ for a number of κ values is plotted in Fig. 5.5. This shows that for $\kappa = 0$, which corresponds to the case of training the RC with data only from $\mathcal{L}$, roughly 50% of the magnitude of $\mathbf{W}^{(2)}_{out}$ element values are greater than 0.08. However, when 20% of the total amount of training data is taken from $\mathcal{L}'$, i.e. $\kappa = 0.2$, then 50% of the magnitude of $\mathbf{W}^{(2)}_{out}$ element values are less than 0.03. This means that by including only a modest amount of data from $\mathcal{L}'$ during the training, this reduces the magnitude of $\mathbf{W}^{(2)}_{out}$ element values by nearly a factor of 3. This trend continues as κ is further increased, and for $\kappa = 0.48$, which corresponds to a slight imbalance in the amount of training data more

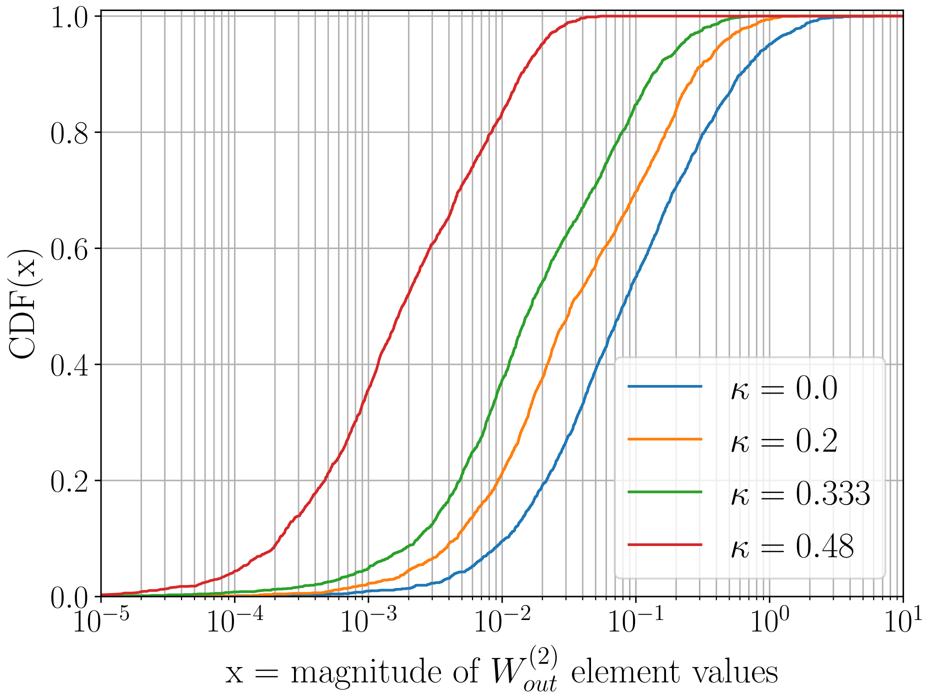


Figure 5.5: Cumulative density function of the magnitude of $\mathbf{W}_{out}^{(2)}$ element values at particular κ values.

than 50% of $\mathbf{W}_{out}^{(2)}$ elements have a magnitude of less than 0.002. While $\mathbf{W}_{out}^{(2)}$ completely vanishes only when $\kappa = 0.5$, it is still strongly suppressed even when there is an imbalance in the amount of training data.

Chaotic Itinerancy

From an attractor reconstruction point of view, an interesting sequence of events is found to occur when increasing ρ past the point of where multifunctionality can be achieved. The dynamics of the RC remains relatively consistent when it is trained to reconstruct a coexistence of $\mathcal{L}$ and $\mathcal{L}'$ from the case shown in Fig. 5.2a with $\rho = 0.2$ up to $\rho = 1.43$ shown in Fig. 5.6. However, by increasing ρ from the value which was used to train the RC, in this case $\rho = 1.43$, to 1.44, Fig. 5.6 shows that $\hat{\mathcal{L}}$ and $\hat{\mathcal{L}}'$ begin to occupy slightly larger regions of $\mathbb{P}$ as the state of the RC takes wider excursions about the extremities of the butterfly attractor's wings. For $\rho = 1.45$ there

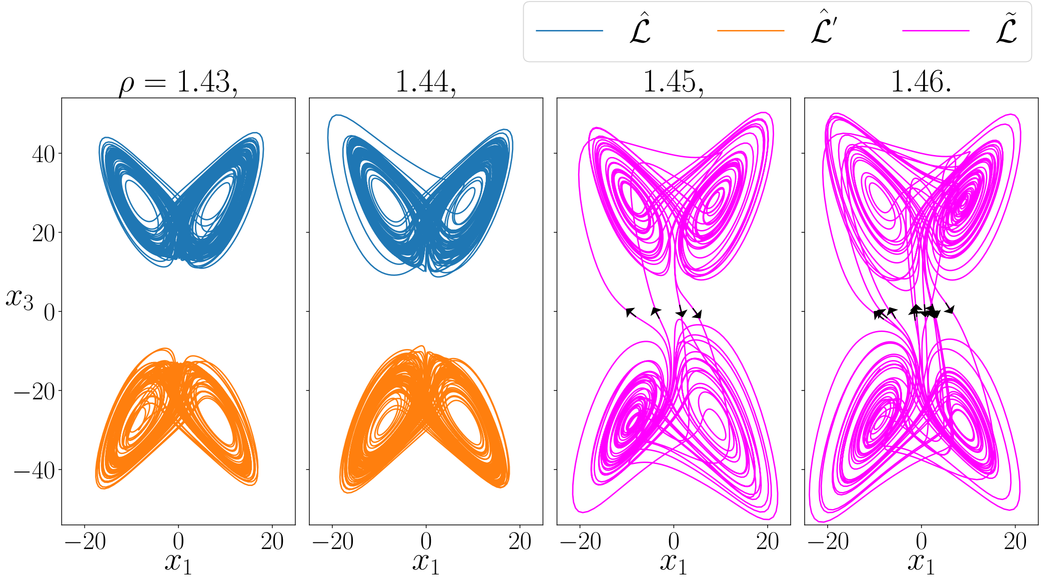


Figure 5.6: Illustrating the changes with respect to ρ in RC's dynamics from exhibiting multifunctionality to chaotic itinerancy.

is a drastic change in the RC's dynamics as it is no longer multifunctional and exhibits the dynamical phenomenon known as 'chaotic itinerancy', which was previously described in Sec. 3.4.4. Furthermore, Fig. 5.6 also shows that the number of transitions between the attractor ruins of $\hat{\mathcal{L}}$ and $\hat{\mathcal{L}}'$ on the chaotically itinerant attractor, $\tilde{\mathcal{L}}$, is seen to increase as ρ is also increased.

As the focus of the present Chapter is on investigating the relationship between symmetry and multifunctionality, the connection between multifunctionality and chaotic itinerancy is further explored in Chapter 8 where it is used to model transitions which take place in the epileptic brain.

5.3.2 Multifunctionality with overlapping mirrored attractors

Symmetry in the RC equations can oftentimes lead to catastrophic consequences once the training data intersects with its mirrored version and this symmetry induced phenomena was discussed in detail by Herteux and Räth in [77] and previously outlined in Sec. 5.1.1. In the present section, these problems are considered from the perspective of multifunctionality.

The behaviour of $\mathbf{W}_{out}^{(2)}$ is investigated in two different cases of overlapping attractors. In the first case, the RC is trained to reconstruct a coexistence of $\mathcal{L}$ and $\mathcal{L}'$ when sharing mutual regions of state space. In the second case, the RC is trained to reconstruct two pairs of overlapping mirrored-Lorenz attractors.

In Sec. 3.2.2 it is stated that a multifunctional RC creates a multistability of attractors in $\mathbb{S}$ such that when these attractors are projected to $\mathbb{P}$ using $\mathbf{W}_{out}$ they resemble the target behaviour. By introducing overlapping regions between the different sources of input data, the importance of the above statement comes to the fore and the results presented in this section help to provide a greater perspective on what needs to happen in $\mathbb{S}$ in order for a RC to achieve multifunctionality. In Chapter 6 the issues relating to overlapping training data are explored in much greater detail.

One pair of overlapping mirrored attractors

In order to produce the overlapping regions in the training data, the location of both $\mathcal{L}$ and $\mathcal{L}'$ are simultaneously shifted along the x_3-direction by a factor ζ in order to preserve the symmetry. In other words, the input data to train the RC to exhibit multifunctionality is written as $\boldsymbol{u}_{\mathcal{L}_\zeta}(t) = (x_1(t), x_2(t), x_3(t) + \zeta)^T$ and $\boldsymbol{u}_{\mathcal{L}'_\zeta}(t) = -\boldsymbol{u}_{\mathcal{L}_\zeta}(t)$. The result of this for $\zeta = -31$ is shown in Figs. 5.7a-5.7b. This shows that the RC is able to reconstruct the overlapping Lorenz attractors. The particular $\mathbf{W}_{out}$ matrix that is used to achieve multifunctionality is shown in Fig. 5.7b which illustrates that even in the case when the attractors are overlapping, the symmetry in the training data eliminates any contribution of $\mathbf{W}_{out}^{(2)}$ to the RC dynamics.

The conditions on which $\mathbf{W}_{out}^{(2)}$ will come into existence are tested by shifting the mirror-attractor along the x_1-axis by a factor χ like in Sec. 5.2.2. In this case the RC is now trained with $\boldsymbol{u}_{\mathcal{L}_\zeta}(t)$ as before and $\boldsymbol{u}_{\mathcal{L}'_{\chi,\zeta}}(t) = (-x_1(t) - \chi, -x_2(t), -x_3(t) - \zeta)^T$. In Figs. 5.7c-5.7d the result of the training for $\chi = 8$ and $\zeta = -31$ is shown. Figs. 5.7c-5.7d precisely illustrates what the analytical work in Sec. 5.2 predicts. Once the symmetry in the training data is broken then $\mathbf{W}_{out}^{(2)}$ exists, even if there is an overlap between the different trajectories described by the training data.

Two pairs of overlapping mirrored attractors

There are many ways to realise the extent of the analytical results in Sec. 5.2 in a numerical setting. As long as the training data satisfies the conditions in Eqs. (5.3)-(5.4), then, in general, from Eq. (5.16) 'symmetry kills the square' when training the RC to reconstruct any number of pairs of mirrored-attractors. In order to illustrate this scenario in a numerical experiment, the blending technique from Sec. 3.2.1 is adapted to train a RC to achieve multifunctionality with more than two attractors. Instead of using the blending parameter α in the matrix concatenation step, the training procedure presented in [77] is used. In this approach, the resultant $\mathbf{X}_C$ and $\mathbf{Y}_C$ matrices, which are used in the Eq. (2.16), are a concatenation of all the individual RC and input data training matrices respectively.

The behaviour of $\mathbf{W}_{out}^{(2)}$ is now analysed when training the RC to reconstruct two pairs of mirrored Lorenz attractors with overlapping regions between all four attractors. Data from the chaotic Lorenz attractor $\mathcal{L}$ is used to generate the training input to the RC for these four attractors. Using the shifting parameters, χ and ζ, the input from the four Lorenz attractors is written as, $\boldsymbol{u}_{\mathcal{L}_1} = (x(t)+\chi, y(t), z(t)+\zeta)^T$, $\boldsymbol{u}_{\mathcal{L}_1'} = -\boldsymbol{u}_{\mathcal{L}_1}$, $\boldsymbol{u}_{\mathcal{L}_2} = (-x(t)-\chi, -y(t), z(t)+\zeta)^T$, and $\boldsymbol{u}_{\mathcal{L}_2'} = -\boldsymbol{u}_{\mathcal{L}_2}$. The result of training the RC to reconstruct a coexistence of $\mathcal{L}_1$, $\mathcal{L}_2$, and their mirrored counterparts for $\chi = -15$ and $\zeta = -40$ is illustrated in Figs. 5.8a-5.8b. Here it can be seen that the RC exhibits multifunctionality by successfully reconstructing each of the four Lorenz attractors. The analytical results from Sec. 5.2 continue to hold as Fig. 5.8b demonstrates that $\mathbf{W}_{out}^{(2)}$ is prohibited from contributing to the prediction due to the symmetry between each individual pair of the mirrored Lorenz attractors in the training data.

To break the symmetry the location of only $\mathcal{L}_1'$ is shifted in the positive x_1 and x_3 direction. Figs. 5.8c-5.8d show that despite the symmetry being present for $\mathcal{L}_2$ and $\mathcal{L}_2'$, it is sufficient to only break the symmetry between $\mathcal{L}_1$

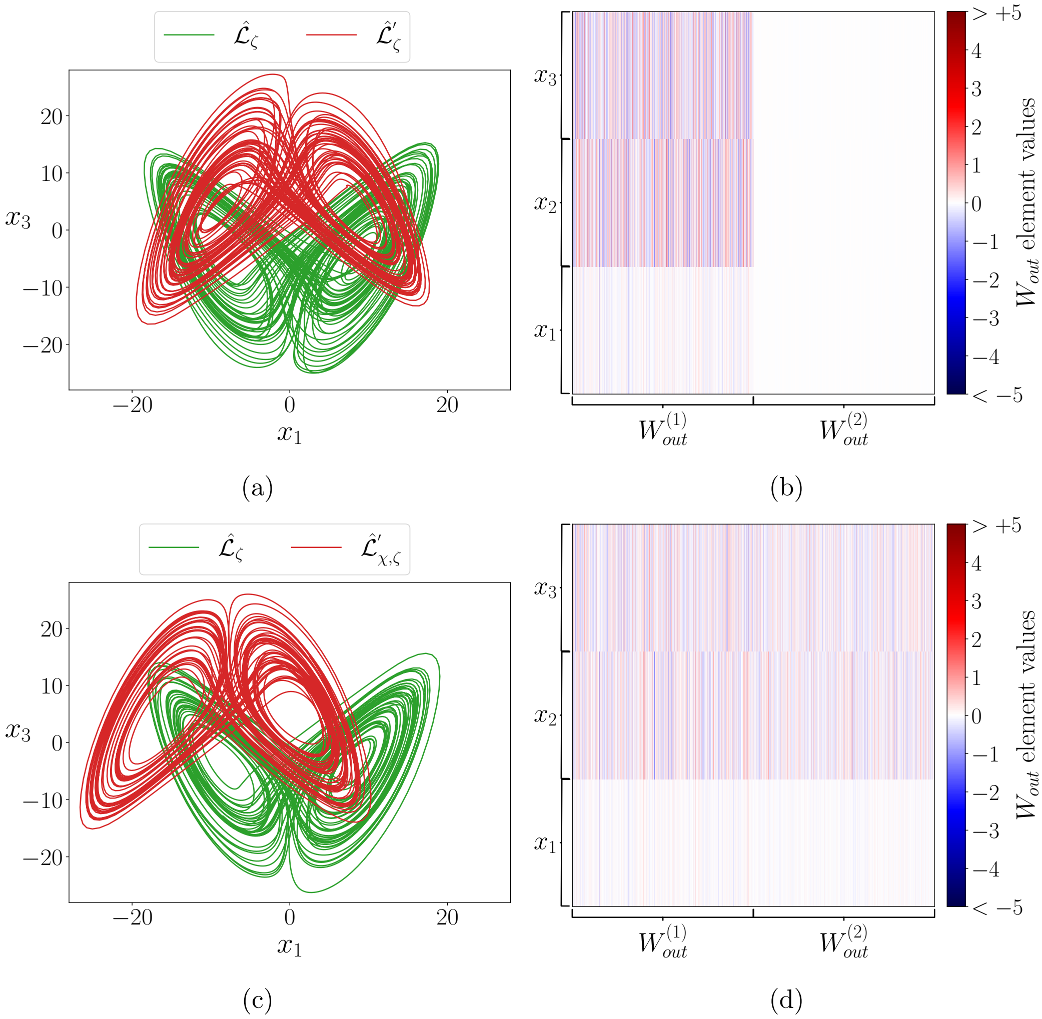


Figure 5.7: Illustrating the behaviour of $\mathbf{W}^{(2)}_{out}$ when the RC is trained to exhibit multifunctionality in cases of overlapping mirror-attractors. (a) Reconstruction of the overlapping $\mathcal{L}_\zeta$ and its mirror-attractor $\mathcal{L}'_\zeta$ and in (b) plot of the corresponding trained $\mathbf{W}_{out}$ matrix. (c) Reconstruction of $\mathcal{L}_\zeta$ and the slightly shifted $\mathcal{L}'_{\chi,\zeta}$ and in (d) the corresponding trained $\mathbf{W}_{out}$ matrix. Description of $\mathbf{W}_{out}$ same as in Fig. 5.2.

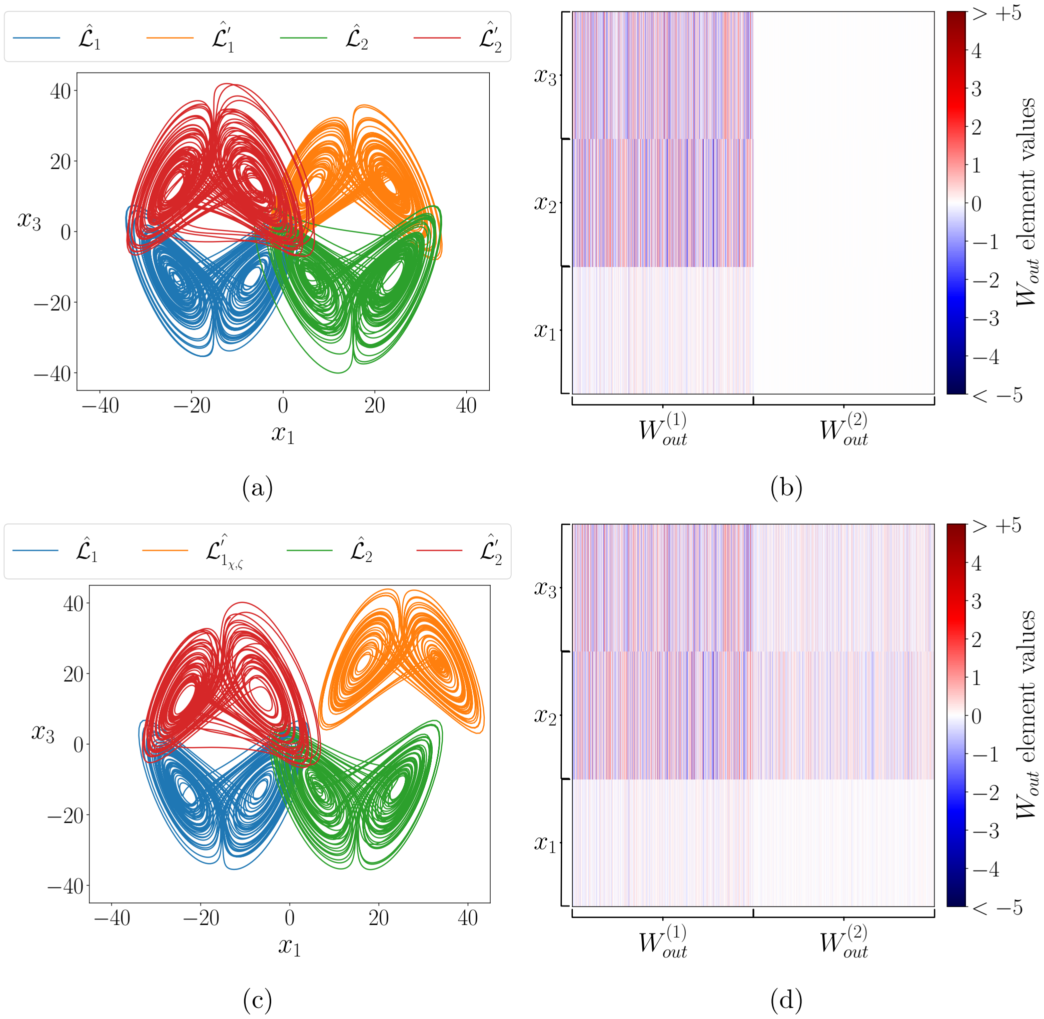


Figure 5.8: Illustrating the behaviour of $\mathbf{W}_{out}^{(2)}$ when the RC is trained to exhibit multifunctionality in cases of two pairs of overlapping mirror-attractors. (a) Reconstruction of the overlapping $\mathcal{L}_1$, $\mathcal{L}_1'$, $\mathcal{L}_2$, and $\mathcal{L}_2'$ and in (b) plot of the corresponding trained $\mathbf{W}_{out}$ matrix. (c) Reconstruction of overlapping $\mathcal{L}_1$, $\mathcal{L}_1'$, $\mathcal{L}_2$, and the slightly shifted $\mathcal{L}_2'$ and in (d) the corresponding trained $\mathbf{W}_{out}$ matrix. Description of $\mathbf{W}_{out}$ same as in Fig. 5.2.

and $\mathcal{L}_1'$ in order for $\mathbf{W}_{out}^{(2)}$ to come into existence.

ρ and overlapping training data

As seen from Table 4.1, the results presented in this section are significantly dependent on an appropriate choice of the spectral radius, ρ, in particular, once there is an overlap in the training data. The state of the RC must have a sufficient knowledge of its previous states in order to avoid spontaneously switching over to any of the other attractors when approaching a region of overlap.

By increasing the value of ρ from 0.2 in Sec. 5.3.1 to 1.7 in this current section, the RC was then able to distinguish between the different input data sources in order to exhibit multifunctionality. When increasing the amount of overlap between the attractors, the RC required larger values of ρ in order to achieve multifunctionality. On the other hand, if ρ is too large then there is a danger that the RC can be pushed beyond the edge of chaos and exhibit dynamical phenomena like chaotic itinerancy as shown in Fig. 5.6 or if pushed too far then the projected dynamics will bear no resemblance to either attractor.

An investigation of the relationship between ρ, multifunctionality, and overlapping data is the basis of the following Chapter.

5.3.3 $\mathbf{W}_{out}^{(2)}$ reappears at large ρ

In the numerical experiments presented so far in the present section, the appearance of $\mathbf{W}_{out}^{(2)}$ has been observed only when both Eqs. (5.3)-(5.4) are violated. In this section, it is shown that $\mathbf{W}_{out}^{(2)}$ can also come into existence in the case where Eq. (5.3) holds but Eq. (5.4) does not at critical values of the spectral radius, ρ.

The remaining case identified in the analytical results in Sec. 5.2.2, where a single source of input training data is already anti-symmetric as described by Eq. (5.17) is now explored numerically using the following source of periodic

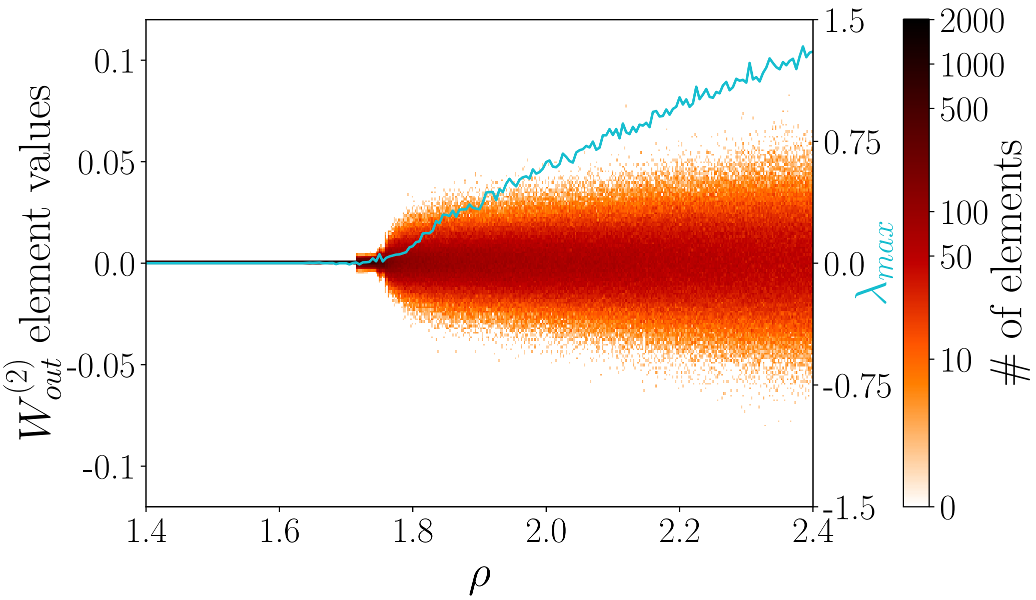


Figure 5.9: Histograms of $\mathbf{W}_{out}^{(2)}$ as a function of ρ. Also plotted is the largest Lyapunov exponent of the predicting RC in Eq. (2.19), λ_{max}, as a function of ρ

training data,

$$\boldsymbol{u}_{\mathcal{C}}(t) = \begin{pmatrix} 5\cos t \\ 5\sin t \end{pmatrix}, \tag{5.20}$$

which describes an orbit on a circle of radius 5 rotating anti-clockwise at constant angular velocity with period $T = 2\pi$. This $\boldsymbol{u}_{\mathcal{C}}(t)$ has the symmetry described in Eq. (5.17). Therefore, when $\boldsymbol{u}_{\mathcal{C}}(t) = (x_1(t), x_2(t))^T$, with $x_1(t)$ and $x_2(t)$ redefined for the case at hand, is used to drive the RC in Eq. (2.10) and the state of the RC fulfils the condition in Eq. (5.18) then Eqs. (5.3)-(5.4) are satisfied so $\mathbf{W}_{out}^{(2)}$ is prevented from coming into existence.

Tuning the spectral radius, ρ, is crucial to a successful training outcome. In this section it is found that if ρ is too large then Eq. (5.18) does not hold despite Eq. (5.17) being fulfilled. The behaviour of $\mathbf{W}_{out}^{(2)}$ is now explored when the RC is trained with $\boldsymbol{u}_{\mathcal{C}}(t)$ as input to Eq. (2.10) for $\rho \in [1.4, 2.4]$.

Fig. 5.9 shows a histogram of the $\mathbf{W}_{out}^{(2)}$ matrix elements as a function of ρ. The darker the colour each point is in Fig. 5.9, the more there are elements of $\mathbf{W}_{out}^{(2)}$ at a given value. Also included in Fig. 5.9 is the largest Lyapunov

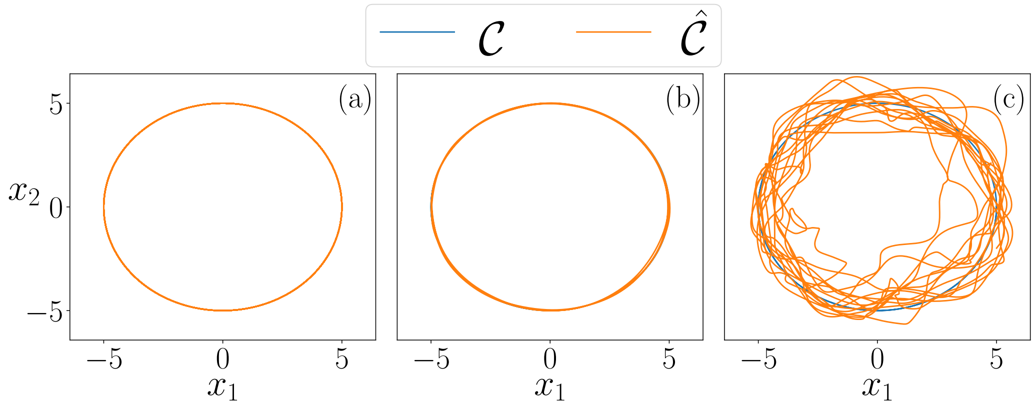


Figure 5.10: Prediction of the circular trajectory $\mathcal{C}$. (a) $\rho = 1.6$, (b) $\rho = 1.75$, (c) $\rho = 2.2$.

exponent of the predicting RC in Eq. (2.19), λ_{max}, as a function of ρ. λ_{max} is computed using the method outlined in Appendix C.1. For $\rho < \rho_s \approx 1.715$ Fig. 5.9 shows that all elements of $\mathbf{W}_{out}^{(2)}$ are 0 and $\lambda_{max} = 0$, indicating that the RC obeys the symmetry conditions specified in Sec. 5.2.2. However, for $\rho_s < \rho < \rho_c = 1.78$ a small number of $\mathbf{W}_{out}^{(2)}$ elements are now nonzero. At the same time $\lambda_{max} \approx 0$ within this range of ρ values. For $\rho > \rho_c$ the RC goes beyond the edge of chaos and as λ_{max} increases from 0 and the elements of $\mathbf{W}_{out}^{(2)}$ begin to grow much faster.

The RC's predicted trajectory on $\mathcal{C}$ at each of the above highlighted stages, $\rho < \rho_s$, $\rho_s < \rho < \rho_c$, and $\rho > \rho_c$ are shown in Fig. 5.10. Here, it can be seen that for increasing values of ρ, the reconstruction of $\mathcal{C}$ goes from good, to bad, to ugly. For $\rho = 1.6$ the RC can successfully reconstruct trajectories on $\mathcal{C}$. However, when $\rho = 1.75$ the reconstruction of $\mathcal{C}$ as viewed in $\mathbb{P}$ is neither chaotic nor symmetric as the prediction begins to breakdown and a mismatch between the RC's dynamics and the ground truth occurs. The RC fails to reconstruct trajectories on the cycle $\mathcal{C}$ for $\rho = 2.2$.

5.4 Conclusions

Breaking the symmetry of certain RC formulations is necessary in order to mitigate the influence of 'mirror-attractors' on the RC's learning capacity [77]. The main focus of this Chapter is to investigate the role of the 'squaring technique' described in Eq. (2.13) and in terms of its relationship with symmetry, mirror-attractors, and multifunctionality. When using Eq. (2.13) in the training of the RC described in Eq. (2.10), a 'square readout matrix', $\mathbf{W}_{out}^{(2)}$, is created which breaks the symmetry in the RC's readout layer and destroys the mirror-attractors as depicted in Fig. 5.1. Throughout this Chapter, the opposite scenario is considered as the behaviour of $\mathbf{W}_{out}^{(2)}$ is explored in the case where there are already certain symmetries present in the training data.

In Sec. 5.2, it is proven analytically that when the RC is trained to reconstruct anti-symmetric training data using the ridge regression technique described in Eq. (2.16), it then follows that $\mathbf{W}_{out}^{(2)}$ must vanish. Examples of this are identified in Eqs. (5.11), (5.16), and (5.17) of anti-symmetric training data which can under certain circumstances 'kill the square' in the readout.

In Sec. 5.3, the scenarios mentioned above are investigated in a numerical setting. To investigate the conditions imposed by Eq. (5.11), the behaviour of $\mathbf{W}_{out}^{(2)}$ is examined when the RC is trained to reconstruct a coexistence of the Lorenz attractor, $\mathcal{L}$, and its corresponding mirror-attractor, $\mathcal{L}'$. Fig. 5.2 shows that when the location of one attractor is slightly shifted, which thereby conflicts with Eqs. (5.3)-(5.4), and only then does $\mathbf{W}_{out}^{(2)}$ appear. Furthermore, the results in Figs.5.3-5.5 demonstrate that the conditions defined in Eqs. (5.3)-(5.4) are robust to small changes in both χ from 0 and κ from 0.5 as the majority of $\mathbf{W}_{out}^{(2)}$ element values remain relatively close to 0. However, the more χ and κ are varied from the above values, the larger the elements of $\mathbf{W}_{out}^{(2)}$ become.

In Fig. 5.6, it is shown that at large values of ρ the RC transitions from multifunctionality to exhibiting chaotic itinerancy as both $\hat{\mathcal{L}}$ and $\hat{\mathcal{L}}'$ have become connected and now exist as one global attractor, $\tilde{\mathcal{L}}$, consisting of quasi-attractors which resemble the previously stable $\hat{\mathcal{L}}$ and $\hat{\mathcal{L}}'$ that the state of the RC now transitions between.

The results illustrated in Fig. 5.7 show that Eq. (5.11) holds even with the added complication of overlapping regions between the pair of mirrored-Lorenz attractors. This result demonstrates that multifunctionality can be used to overcome the harmful properties of symmetry as described in [77].

To explore the square killing scenario described by Eq. (5.16) in a numerical setting, the RC is trained to reconstruct two pairs of mirrored Lorenz attractors in Sec. 5.3.2. Fig. 5.8 demonstrates that it is sufficient to break the symmetry between one pair of mirrored-Lorenz attractors in order for $\mathbf{W}_{out}^{(2)}$ to come into existence.

In Sec. 5.3.3 the case of a single anti-symmetric trajectory, as described in Eq. (5.17), is explored numerically using a circular orbit described by Eq. (5.20). Here, it is found that when the RC is successfully able to reconstruct a trajectory on this circular orbit there is no contribution from $\mathbf{W}_{out}^{(2)}$. However, once ρ is too large then the prediction breaks down as the RC begins to exhibit chaos and $\mathbf{W}_{out}^{(2)}$ emerges.

Overall, this Chapter demonstrates the benefits of studying the effects of symmetries in a RC setup and serves as a route to better understand the fundamentals of this ML paradigm. The two-armed approach of analytical results together with numerical experiments provides a much greater insight towards the relationship between symmetry and multifunctionality in a RC. The inclusion of exotic cases like overlapping regions between attractors in $\mathbb{P}$ and achieving multifunctionality with four attractors further increase the robustness of the results discussed in this Chapter and showcases some of the interesting dynamics a RC can be trained to simultaneously reconstruct.

Chapter 6

Seeing Double

The results presented in Chapters 4 and 5 provide a glimpse at some of the interesting multistabilities that multifunctional RCs are capable of possessing. These include examples of RCs reconstructing a coexistence of attractors from the same dynamical system, different dynamical systems, and even when there are overlapping regions in state space between the attractors.

The focus of this Chapter is to explore the relationship between overlapping training data and multifunctionality. It is through examining the dynamics of the RC when solving the 'seeing double' problem that a fundamental understanding of multifunctionality is obtained.

The conceptual framework in Sec. 3.2.1 which describes the *mise-en-scène* on how RCs achieve multifunctionality is based on the knowledge gained from the analytical and numerical results presented in this Chapter.

6.1 Issues with Overlapping Data

From both a philosophical and practical perspective, a natural question to consider when training a RC to achieve multifunctionality is, what happens when there are overlapping regions in state space between the different attractors that the RC is required to reconstruct a coexistence of?

On one hand, no autonomous dynamical system is allowed to have coexisting attractors which share common points of its state space. On the other hand, a dynamical system in the form of multifunctional RC is capable of reconstructing a coexistence of attractors even if the attractors are overlapping in state space. This apparent confliction imbues further motivation to clarify how, in general, RCs become multifunctional.

The results presented in Sec. 6.2 provide additional context behind the issues highlighted above. However there is a limit to the level of insight which can be gained from these results. This subsequently motivates the need for the reductionist approach taken in Sec. 6.3 where the 'seeing double' problem sheds further light on the relationship between multifunctionality and overlapping training data.

6.2 Reconstructing Lorenz and Halvorsen

The task outlined in Sec. 6.2.1 is used to illustrate issues involving overlapping training data which arise when training different types of RCs, namely the continuous-time (CT), leaky-integrator (LI), and next generation (NG) RCs specified in Eqs. (2.10), (2.21), and (2.25) respectively.

6.2.1 Outline of task

Each of the CT, LI, and NG RCs are trained to reconstruct a coexistence of the chaotic Lorenz attractor, $\mathcal{L}$ and the chaotic Halvorsen attractor, $\mathcal{H}$, solutions of which are computed from the Halvorsen system described by the following set of equations,

$$\begin{aligned} \dot{x}_1 &= -1.3x_1 - 4x_2 - 4x_3 - x_2^2, \\ \dot{x}_2 &= -1.3x_2 - 4x_3 - 4x_1 - x_3^2, \\ \dot{x}_3 &= -1.3x_3 - 4x_1 - 4x_2 - x_1^2. \end{aligned} \tag{6.1}$$

This pairing of attractors was studied in [77] when investigating the relationship between symmetry and mirror-attractors.

The training data is obtained by generating a trajectory on $\mathcal{L}$ and $\mathcal{H}$ from Eqs. (2.28) and (6.1) using the 4^{th} order Runge-Kutta method with time step $\tau = 0.01$. The input training data sets, $\mathbf{Y}_{\mathcal{L}}$ and $\mathbf{Y}_{\mathcal{H}}$, which are used to drive a response from each RC, describe a trajectory on $\mathcal{L}$ and $\mathcal{H}$ which has been normalised such that the furthest point from the origin on each attractor is

RC	ρ	σ	γ	η	β	O	k	s	t_l	t_w
CT	1.6	5	7	n/a	10^2	n/a	n/a	n/a	100	n/a
LI	0.9	1.2	n/a	0.2	10^{-3}	n/a	n/a	n/a	100	n/a
NG	n/a	n/a	n/a	n/a	3×10^{-5}	$1, 2, 3, 4, 5$	3	2	n/a	6

Table 6.1: Hyperparameters and time parameters used to generate the results displayed in Fig. 6.1. $t_l = t_{\text{listen}}$ and $t_w = t_{warm}$.

less than 1. The resulting 'normalised attractors' are from now on referred to as $\mathcal{L}$ and $\mathcal{H}$ during this section. Both normalised attractor data sets are moved equidistantly from the origin in opposite directions along the x_3-axis by a factor, δx_3, known as the 'shift parameter' in order to examine how each RC performs once the attractors are moved closer together or further apart and begin to overlap.

6.2.2 Training parameters

The training parameters used in each RC setup were chosen using a grid search method for the above task and are specified in Table 6.1. The adjacency matrix for both 'traditional' RCs (the CT and LI-RCs) was designed using $N = 1000$ with $P = 0.05$ and $P = 0.012$ for the CT and LI-RC setups respectively. In each RC setup, the same blending technique from Sec. 3.2.1 was used to construct the respective training data matrices and corresponding $\mathbf{W}_{out}$ with $\alpha = 0.5$ and $t_{\text{train}} = 200$.

6.2.3 Comparing CT, LI, and NG-RCs

Each of the RCs' reconstruction of $\mathcal{L}$ and $\mathcal{H}$ is illustrated in Fig. 6.1 for certain values of $\delta x_3 \in [0, 2]$. Here it can be seen that all RCs reconstruct a coexistence of $\mathcal{L}$ and $\mathcal{H}$ for $0.75 \leq \delta x_3 \leq 1.25$.

For the set of training parameters specified in Table 6.1, Fig. 6.1 shows that no RC is capable of reconstructing a coexistence of $\mathcal{L}$ and $\mathcal{H}$ for all δx_3's and in particular for $\delta x_3 = 0$.

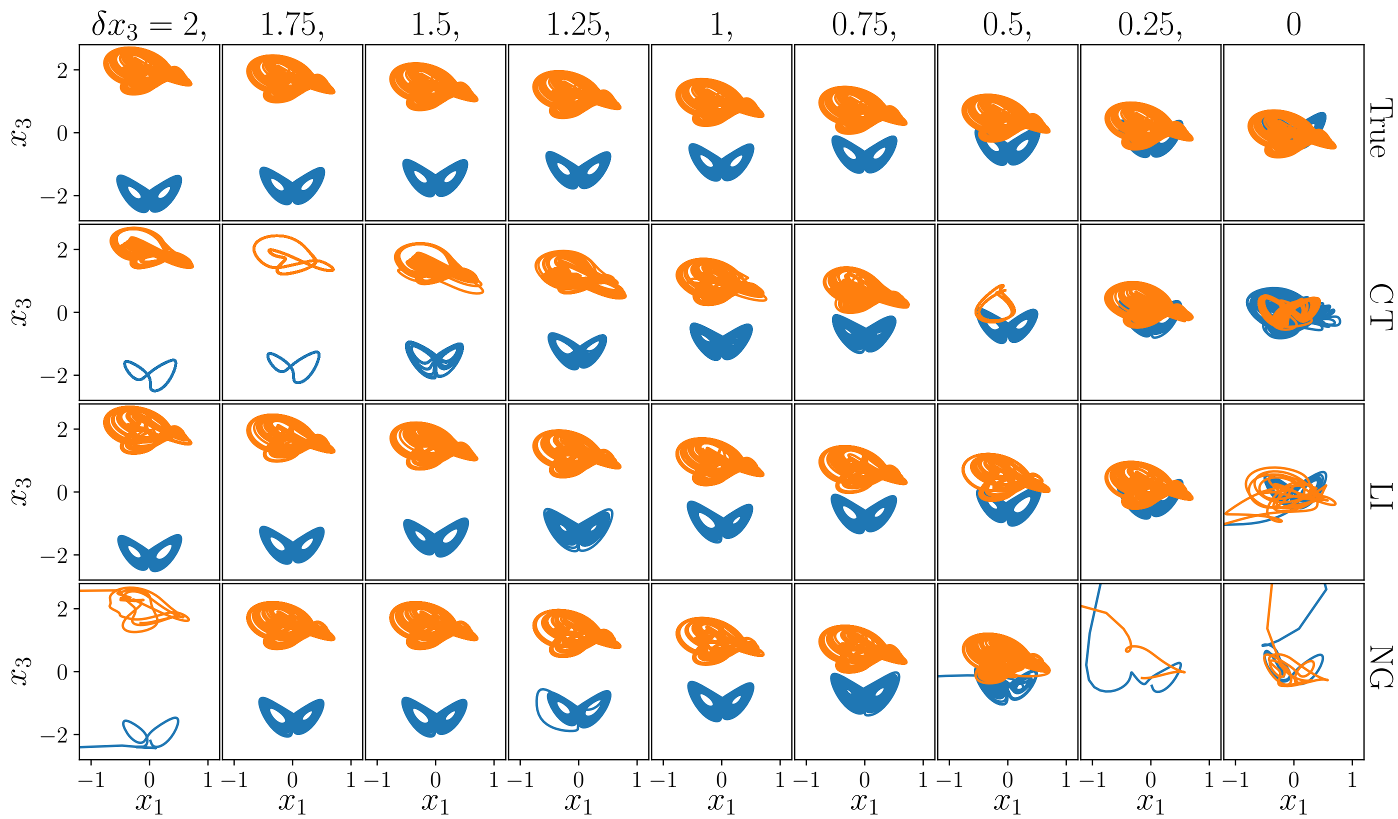


Figure 6.1: Comparison of CT (continuous-time), LI (leaky-integrator), NG (next-generation) RC's ability to achieve multifunctionality when trained to reconstruct a coexistence of $\mathcal{L}$ (blue) and $\mathcal{H}$ (orange) which are separated by a factor of δx_3 away from the origin along the x_3-axis.

Fig. 6.1 highlights that in modes of failure there are significant operational differences between the NG-RC and the 'traditional' CT and LI-RCs. In these examples, when either $\mathcal{L}$ or $\mathcal{H}$ cannot be reconstructed, the state of the NG-RC tends to infinity whereas the CT and LI-RCs prediction decay to some other attractor.

In addition, there are some noticeable differences in the performance of each RC as the attractors are brought closer together and further apart from $\delta x_3 = 1$.

Decreasing $\boldsymbol{\delta x_3}$ from 1

Fig. 6.1 shows that the NG-RC is unable to reconstruct both attractors in the case of overlapping data, the CT and LI-RCs are able to provide a reasonable reconstruction of both $\mathcal{L}$ and $\mathcal{H}$ for $\delta x_3 = 0.25$ where large portions of the attractors are already overlapping.

Furthermore, these results indicate that the relationship between multifunctionality and overlap is not always linear as, for instance, the CT-RC fails to achieve multifunctionality at times when there is less of an overlap between the attractors, i.e., for $\delta x_3 = 0.5$ the CT-RC only reconstructs $\mathcal{L}$ but for $\delta x_3 = 0.25$ both are reconstructed.

Increasing $\boldsymbol{\delta x_3}$ from 1

As $\mathcal{L}$ and $\mathcal{H}$ are moved away from one another, both the CT and NG-RCs fail to achieve multifunctionality with the CT-RC's predicted trajectory on $\mathcal{L}$ and $\mathcal{H}$ decaying to different limit cycles and the NG-RC becoming unstable as its state tends to infinity. The same occurs for the LI-RC as its state tends to limit cycles for larger values of δx_3 although it is not shown in Fig. 6.1.

As a further comment, in other experiments whose results are also not shown in this section, it was found that the CT and NG-RCs achieve multifunctionality for $\delta x_3 > 1.5$ with different training parameters. However, by keeping the parameters fixed, this provides further insight towards the complex relationship between multifunctionality and overlapping training data.

Insight limitations

While the above results shed some light on issues relating to overlapping training data, there is a limit to what can be understood about multifunctionality from this arbitrary choice of attractors, $\mathcal{L}$ and $\mathcal{H}$.

As previously depicted in Fig. 4.7b and discussed in Sec. 4.3.2, for an RC to reconstruct a coexistence of attractors from different dynamical systems is a difficult task in and of itself. Subtleties such as the variation between the nonlinearities and structure of the dynamical systems which produce $\mathcal{L}$ and $\mathcal{H}$ therefore require the RC to become some manifestation of both in order to achieve multifunctionality.

To combat this and mitigate the impact of the complications identified above, the 'seeing double' problem is introduced in the following section as a means to place increased focus on studying the relationship between multifunctionality and overlap.

6.3 The Seeing Double Problem

A numerical experiment called the 'seeing double' problem is introduced in this section. The experiment is designed as a means to systematically induce the issues related to multifunctionality and overlapping training data in a manner which avoids the complications caused by chaos in Sec. 6.2.

The task involves training a RC to reconstruct a coexistence of attractors which follow trajectories on two circular orbits of equal radius that rotate in opposite directions. A parameter is introduced that allows the centres of these cycles to be moved either closer together or further apart in an equidistant manner. When these orbits are overlapping, the RC is therefore required to distinguish between points in space that are common to both cycles in order to exhibit multifunctionality. Further specifics of the experiment are outlined in Sec. 6.3.1.

By virtue of the seeing double problem's simplicity, a greater emphasis can be placed on examining how multifunctionality arises in a RC. At the same time, despite this simplicity, an enormity of elaborate dynamics are encountered and studied in Sec. 6.4.

6.3.1 Numerical experiment setup

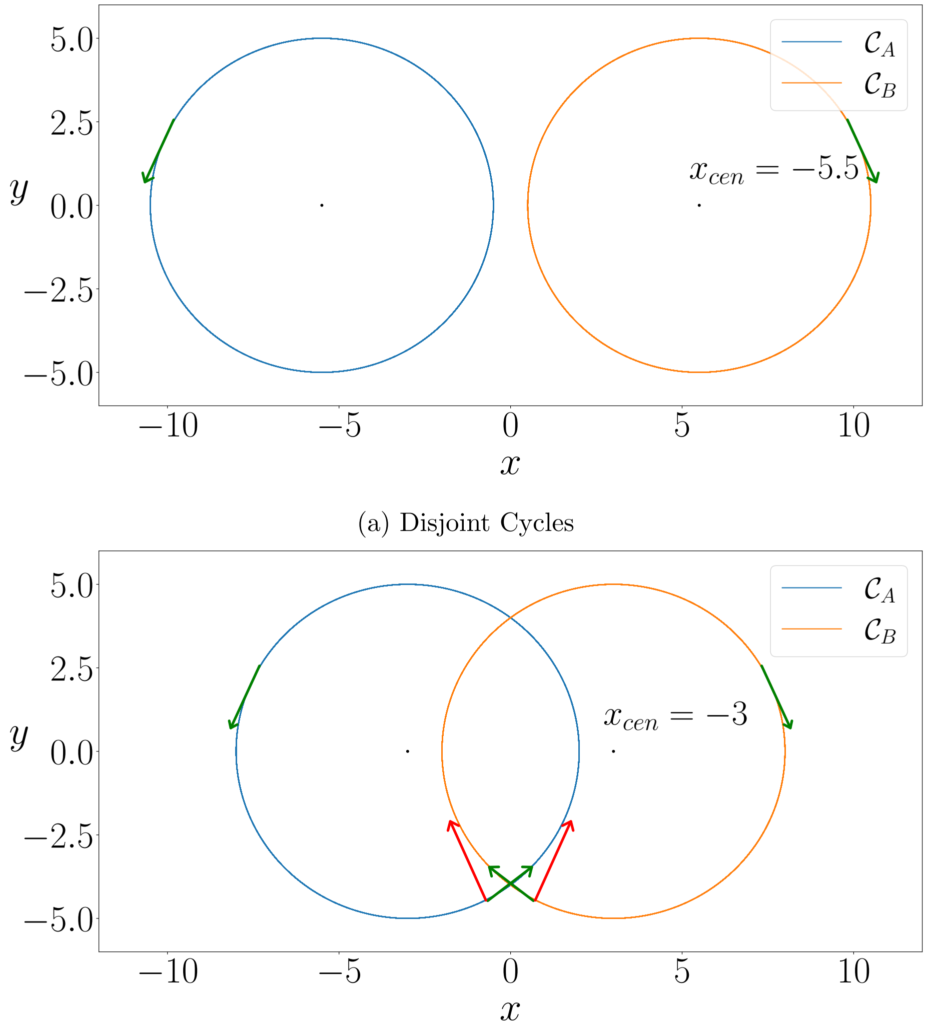
The training data input sequences for the RC are computed via,

$$\boldsymbol{u}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} b_x \cos(t) + x_{cen} \\ b_y \sin(t) + y_{cen} \end{pmatrix}, \tag{6.2}$$

for $t = 0, \tau, 2\tau, \ldots$, using the time-step $\tau = 0.01$ to discretise the data. The resultant $\boldsymbol{u}(t)$ generated from Eq. (6.2) produces time-series data which resembles a trajectory around a circle of radius $b = |b_x| = |b_y|$ and centered at (x_{cen}, y_{cen}). Two sets, $\mathcal{C}_A$ and $\mathcal{C}_B$, are produced using Eq. (6.2) for a given b_x, b_y, x_{cen}, and y_{cen} and the corresponding input time-series which are used to train the RC are denoted by $\boldsymbol{u}_{\mathcal{C}_A}$ and $\boldsymbol{u}_{\mathcal{C}_B}$.

In this Chapter, b_x and b_y are set to $b_x = b_y = 5$ to create $\mathcal{C}_A$ and for $\mathcal{C}_B$, $b_x = -5$ and $b_y = 5$. The centres of $\mathcal{C}_A$ and $\mathcal{C}_B$ are brought closer together by changing the values of (x_{cen}, y_{cen}) to bring about the issues of overlap. In order to simplify the experiment further, $\mathcal{C}_A$ and $\mathcal{C}_B$ are designed to be centered at $(x_{cen}, 0)$ and $(-x_{cen}, 0)$ respectively. Therefore, by changing x_{cen} the centres of these cycles are moved equidistantly along the line $y = 0$. In this case, an overlapping region between $\mathcal{C}_A$ and $\mathcal{C}_B$ exists whenever $|x_{cen}| \leq b = 5$, i.e., $\mathcal{C}_A \cap \mathcal{C}_B \neq \emptyset \ \forall |x_{cen}| \leq 5$. Furthermore, $\mathcal{C}_A$ and $\mathcal{C}_B$ are said to be 'entirely overlapping' when $x_{cen} = 0$. In this extreme case, the only difference between $\mathcal{C}_A$ and $\mathcal{C}_B$ is the direction of rotation on both cycles.

Using the terminology developed in Sec. 3.2.1 regarding the steps involved in achieving multifunctionality, in the seeing double problem, the aim for the RC is to reconstruct a coexistence of two attractors which resemble $\mathcal{C}_A$ and $\mathcal{C}_B$ when projected to $\mathbb{P}$. These projected attractors are denoted by $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ and are projections of the higher-dimensional limit cycles, $\mathcal{A}$ and $\mathcal{B}$ which exist in $\mathbb{S}$.



(a) Disjoint Cycles

(b) Cycles Intersect

Figure 6.2: Illustration of moving the centres of the $\mathcal{C}_A$ and $\mathcal{C}_B$ along the line $y = 0$ with parameter x_{cen}. In (a) orbits do not intersect for $x_{cen} = -5.5$, and in (b) orbits are overlapping when $x_{cen} = -3$. Green arrows indicate a trajectory on these cycles. Red arrows indicate a potential path to failure.

6.3.2 Illustrating issues of overlap

The cycles $\mathcal{C}_A$ and $\mathcal{C}_B$ produced from Eq. (6.2) are depicted in Fig. 6.2 with $\boldsymbol{u}_{\mathcal{C}_A}$ and $\boldsymbol{u}_{\mathcal{C}_B}$ plotted for a given x_{cen}. The green arrows indicate the direction of rotation in which an orbit along these circles evolves upon in time.

Fig. 6.2a shows that the cycles do not overlap when $x_{cen} = -5.5$. However, when $|x_{cen}| \leq 5$, like in Fig. 6.2b for $x_{cen} = -3$, the orbits share similar regions of the (x, y)-space. This presents a challenge for the RC to remain on the correct path when its state approaches the point where these cycles intersect as indicated by the green arrows at the crossing point in Fig. 6.2b. If the RC fails to distinguish between these overlapping cycles then, for example, the state of the RC may move onto a different path, i.e., by following the red arrow and cross over to the other cycle.

Therefore, when there is an overlap between $\mathcal{C}_A$ and $\mathcal{C}_B$ then the RC must have sufficient knowledge of its previous states in order to remain on the correct path when approaching a crossing of the different cycles.

Later in this Chapter, it is found that by changing the spectral radius, ρ, of the RC's internal connections, which as a result tunes the weight given to the RC's previous states, it is possible to overcome the issues of overlap.

In the next Chapter, the extreme case of the seeing double problem where $x_{cen} = 0$ is considered as a benchmark task for testing the limits of multifunctionality across different RC setups. However, these results show that a good choice of ρ for a given random realisation of a RC setup may not work well for all RCs. From further exploration a relationship between multifunctionality, ρ, randomness, and dimensionality is identified.

6.4 Seeing Double with a Multifunctional RC

The focus of this section is to explore the dynamics of the CT-RC (defined in Eq. (2.19)) when it is trained to solve the seeing double problem. The difficulty of the seeing double problem is varied according to changes in x_{cen} as the closer $\mathcal{C}_A$ and $\mathcal{C}_B$ are to one another the more memory is required of the RC and consequently the spectral radius, ρ, plays a more significant role.

N	P	ρ	σ	γ	β	t_{listen}	t_{train}
1000	0.04	$[0.1, 2.5]$	0.2	5	10^{-2}	200	400

Table 6.2: RC design and training parameters used to generate the results displayed in Sec.6.4.

A strong emphasis is placed on how the training is effected by changes in x_{cen} and ρ.

The results in this section are generated by training the CT-RC using the method outlined in Sec. 3.2.1 with $\alpha = 0.5$ and the input data, $\boldsymbol{u}_{\mathcal{C}_A}(t)$ and $\boldsymbol{u}_{\mathcal{C}_B}(t)$ constructed as described in Sec. 6.3.1. The RC training and design parameters are specified in Table 6.2 and the same $\mathbf{M}$ and $\mathbf{W}_{in}$ matrices are used to produce the results presented throughout this Chapter.

6.4.1 Exploring Multifunctionality in (x_{cen}, ρ)-plane

The long-term dynamics of the predicting RC (described in Eq. (2.19)) when initialised with $\hat{\boldsymbol{r}}(0) = \boldsymbol{r}_{\mathcal{C}_A}(t_{\text{listen}})$ and $\hat{\boldsymbol{r}}(0) = \boldsymbol{r}_{\mathcal{C}_B}(t_{\text{listen}})$ are explored, as viewed from $\mathbb{P}$, for a given $x_{cen} \in [-10, 10]$ and $\rho \in [0.1, 2.5]$.

Prediction Analysis

The method which is used to analyse whether the RC solves the seeing double problem for a particular choice of x_{cen} and ρ is similar to the approach used in Sec. 4.3.1 and is outlined as follows.

After the training a number of outcomes are possible. If the training is successful then $\hat{\mathcal{C}}_A \approx \mathcal{C}_A$ and $\hat{\mathcal{C}}_B \approx \mathcal{C}_B$ and the RC's predicted trajectory of either $\mathcal{C}_A$ or $\mathcal{C}_B$ will remain on $\hat{\mathcal{C}}_A$ or $\hat{\mathcal{C}}_B$ in the long term ($t \to \infty$). However, given the nature of the seeing double problem, it is also possible for the predicted trajectory to switch from $\hat{\mathcal{C}}_A$ to $\hat{\mathcal{C}}_B$ or vice-versa if the RC fails to lift the corresponding driving input data, $\boldsymbol{u}_{\mathcal{C}_A}(t)$ and $\boldsymbol{u}_{\mathcal{C}_B}(t)$, into separate regions of $\mathbb{S}$. Furthermore, if the RC fails to reconstruct $\mathcal{C}_A$ or $\mathcal{C}_B$ then the predicted trajectory can, for example, decay to a fixed point, some other limit cycle, or never settle down to periodic behaviour and display evidence of the

RC entering a chaotic regime.

To assess the accuracy of the RC's prediction of a given orbit an error metric known as the 'roundness' is used. The roundness of a given cycle, $\mathcal{C}$, is denoted by $\delta\left(\mathcal{C}\right)$ and is found by calculating the difference between the radii of the maximum and minimum circles needed to enclose and inscribe the cycle. For simplicity, the centres of these maximum and minimum circles are taken to be located at the centre of the given target orbit. From multiple trial runs it was found that if the relative roundness of a given cycle, $\delta_{rel}\left(\mathcal{C}\right) = \delta\left(\mathcal{C}\right)/b$, was $< \delta_{rel}^{+} = 0.25$, then the RC is in general able to successfully reconstruct the desired orbit with a high level of accuracy. This error metric is typically used in the production of circular shaped materials, like plastic tubes or copper pipe, and the formula to calculate the roundness is found in most machining manuals in line with the International Organisation for Standardisation (ISO).

The RC's reconstruction of a given orbit, i.e., the long-term dynamics of the RC when initialised with either $\boldsymbol{r}_{\mathcal{C}_A}(t_{\text{listen}})$ or $\boldsymbol{r}_{\mathcal{C}_B}(t_{\text{listen}})$, is characterised as the particular attractor which the RC's dynamics is on for $t_{\text{predict}} - t^* \leq t \leq t_{\text{predict}}$ where $t_{\text{predict}} = 600$ and $t^* = 40$.

The result of applying the following characterisation procedure is displayed in Figs. 6.3a-6.3b where each point in the (x_{cen}, ρ)-plane is coloured:

1. Blue, if the correct cycle is reconstructed and $\delta_{rel}\left(\mathcal{C}\right) < \delta_{rel}^{+}$.
2. Yellow, if the prediction switches to the other cycle.
3. Purple, if the prediction does not settle down to periodic motion and label this as some attractor, $\mathcal{S}$.
4. Black, if the prediction tends to some other limit cycle or $\delta_{rel}\left(\mathcal{C}\right) \geq \delta_{rel}^{+}$, this attractor is labelled as, LC.
5. Green, if the prediction decays toward some fixed point labelled as FP.

In order to deduce the regions of the (x_{cen}, ρ)-plane in which multifunctionality was achieved, an additional picture is provided in Fig. 6.4 whereby the common blue regions of Figs. 6.3a-6.3b are identified and the larger of the two relative roundness values when assessing the accuracy of the RC's reconstruction of $\mathcal{C}_A$ and $\mathcal{C}_B$, defined as $\delta_{rel}^{max} = \max\left(\delta_{rel}\left(\hat{\mathcal{C}}_A\right), \delta_{rel}\left(\hat{\mathcal{C}}_B\right)\right)$, is plotted for each point in the (x_{cen}, ρ)-plane.

Regions of multifunctionality

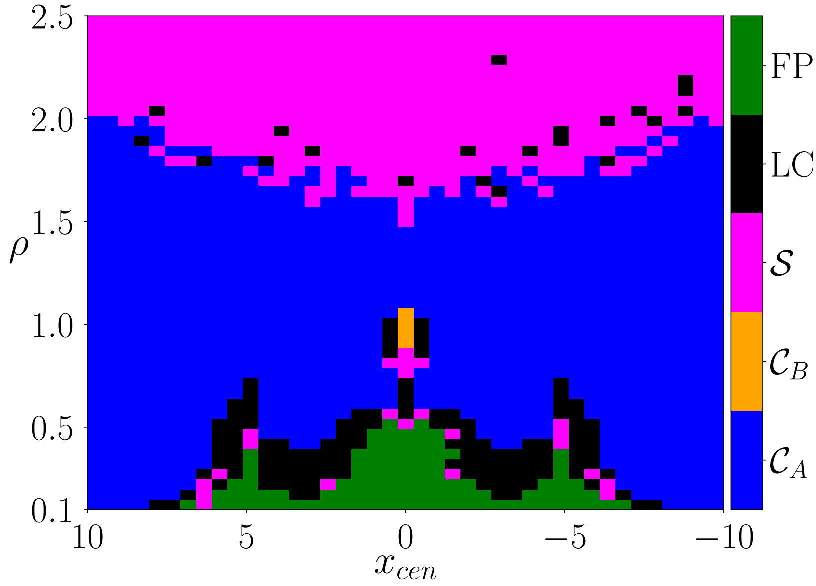
The relationship between overlapping training data, multifunctionality, and ρ becomes evident in Figs. 6.3-6.4 where a 'Goldilocks effect' is found to occur. By this it is meant that if ρ is too small or too large then the RC fails to reconstruct both orbits. When the orbits are furthest apart then the RC is able to achieve multifunctionality even at small values of ρ. However, as the orbits are moved closer together and $\mathcal{C}_A \cap \mathcal{C}_B \neq \emptyset$, then the RC requires a much larger memory in order to distinguish between the cycles and prevent the predicted trajectories from decaying to a fixed point or some other limit cycle. If ρ is too large the RC fails to reconstruct either orbit regardless of the investigated values of x_{cen} and displays evidence of chaos.

Furthermore, it is important to note that a near symmetric set of results occur about $x_{cen} = 0$ in all Figs. 6.3-6.4. Further results on symmetry are discussed later in Sec. 6.4.4.

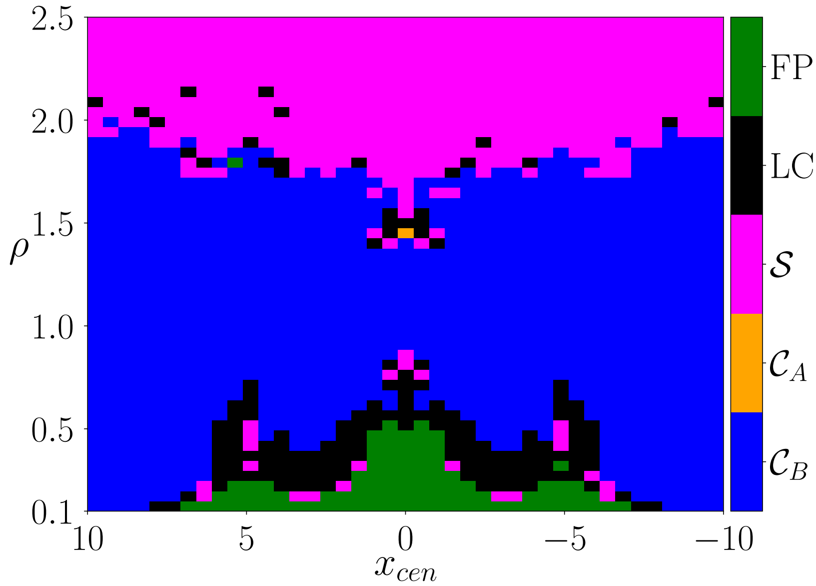
$\mathcal{C}_A$ and $\mathcal{C}_B$ rotating in same direction

To provide a comparison, the same RC (constructed via the parameters in Table 6.2) is trained to reconstruct a coexistence of $\mathcal{C}_A$ and $\mathcal{C}_B$ where trajectories on both cycles are instead rotating in the same direction, i.e., when $b_x = b_y = 5$ for both $\mathcal{C}_A$ and $\mathcal{C}_B$.

The regions of multifunctionality in the (x_{cen}, ρ)-plane are computed using the same procedure which produced Fig. 6.4. The result of this is presented in Fig. 6.5 where a similar phenomenon to Fig. 6.4 is shown to occur, once the overlap is introduced then multifunctionality is achieved only when ρ is sufficiently large. However, the main difference between Fig. 6.5 and Fig. 6.4 is the trivial case for $x_{cen} = 0$ as $\mathcal{C}_A$ and $\mathcal{C}_B$ are the exact same and the RC has no issue reconstructing the dynamics for $\rho \in [0.1, 1.7]$. For $\rho > 2.1$, the RC fails to become multifunctional $\forall x_{cen} \in [-10, 10]$.



(a) $\mathcal{C}_A$ prediction



(b) $\mathcal{C}_B$ prediction

Figure 6.3: Characterising the prediction of (a) $\mathcal{C}_A$ and (b) $\mathcal{C}_B$ in the (x_{cen}, ρ)-plane.

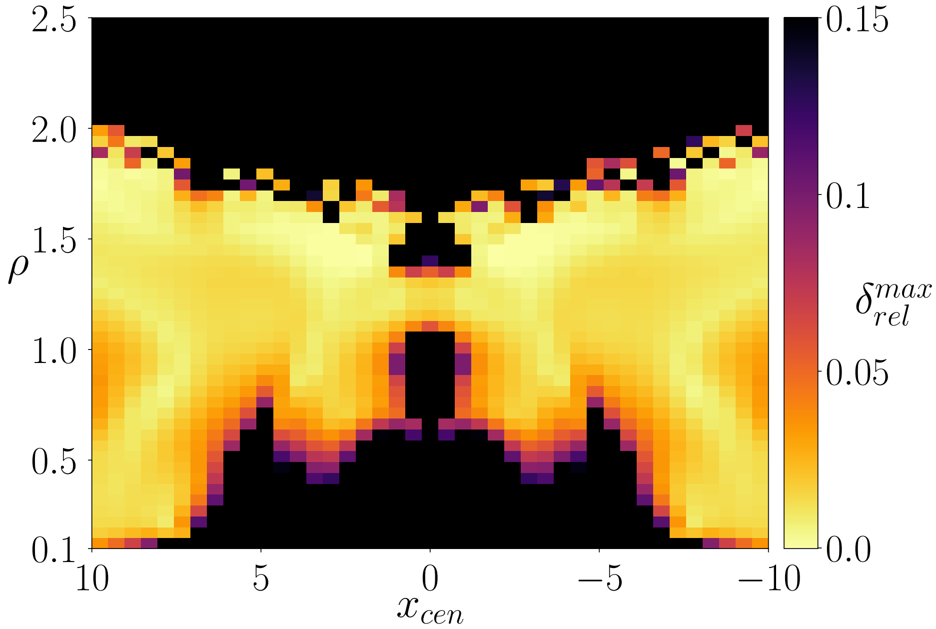


Figure 6.4: Regions of multifunctionality with error δ_{rel}^{max} for $\mathcal{C}_A$ and $\mathcal{C}_B$ rotating in opposite directions.

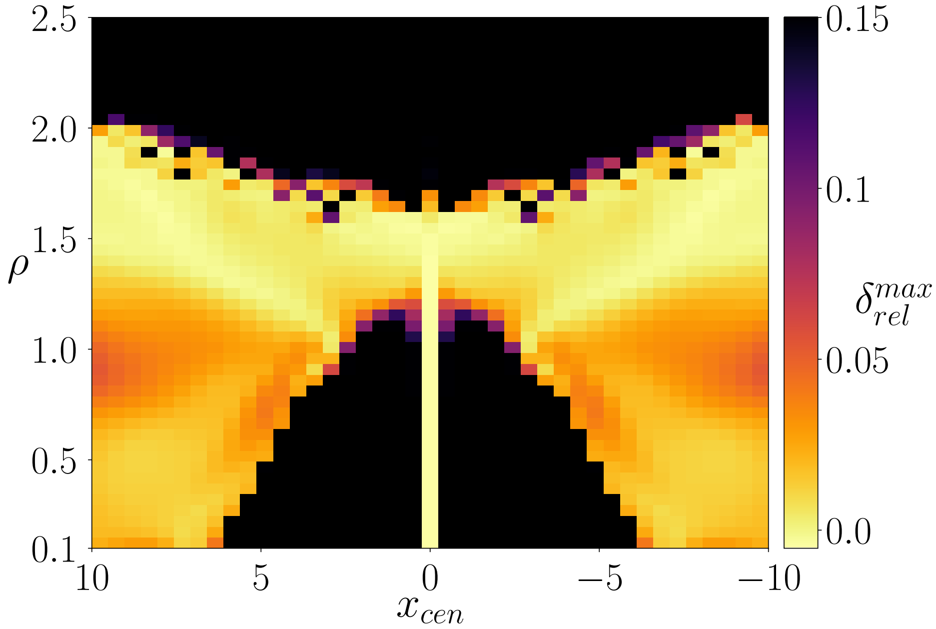


Figure 6.5: Regions of multifunctionality with error δ_{rel}^{max} for $\mathcal{C}_A$ and $\mathcal{C}_B$ rotating in the same direction.

Edge of Chaos

The smallest error, δ_{rel}^{max}, for a given x_{cen} and ρ in Figs. 6.4 and 6.5 occur, in many instances, before the transition from success to failure as the RC's behaviour becomes chaotic for large ρ. This transition is known in the RC and wider research community as the 'edge of chaos' or the 'edge of stability'. This phenomenon has been the focus of many investigations which suggest that RCs achieve optimal computational capacity at the edge of chaos [143–145]. However, Carroll shows in [40] that not all RCs perform best precisely at this transition. Like the results presented in Figs. 6.4 and 6.5, Carroll provides several numerical examples of RCs which perform better on certain problems just prior to crossing the edge of chaos.

As a further comment, Fig. 6.5 shows that, the edge of chaos, which in this instance is also the border between failure and success appears in the same vicinity of ρ values as it does in Fig. 6.4 for $|x_{cen}| \gtrsim 1.5$. As the same RC is used in both cases, this suggests that the location of the edge of chaos may also arise as an intrinsic property of the RC rather than solely being related the nature of the task itself. At the same time, this comment is only speculation and considerable investigation is required in order to shed further light on this.

However, progress has already been made in answering questions of the above nature. As mentioned previously in Sec. 2.3.4, Jiang and Lai conducted many extensive experiments in [78] which provide significant insight towards assessing the role of ρ in changes to the performance of the original ESN design described in Eq. (2.22) when training 100 random realisations of this type of RC to reconstruct the dynamics of several different systems. The central result from Jiang and Lai's experiments is that, for a given RC, there exists an interval of ρ values where optimal or near-optimal performance is achieved. A similar phenomenon is seen to occur in Figs. 6.4 and 6.5 which compliments the work of Jiang and Lai and also offers additional information as Figs. 6.4 and 6.5 both show that as the difficulty of the task involved is varied (by changing x_{cen}) then the width of these intervals change accordingly. Additional questions involving the role of ρ, randomness, and RC dimension are explored in Chapter 7.

6.4.2 Achieving multifunctionality when $x_{cen} = 0$

From a dynamical systems perspective it is remarkable that a RC can reconstruct a coexistence of $\mathcal{C}_A$ and $\mathcal{C}_B$ when these share similar regions of state space. However, the most striking result from Fig. 6.4 is that the RC is able to reconstruct a coexistence of these orbits even when they are completely overlapping, i.e., when $x_{cen} = 0$. In this extreme scenario, the RC achieves multifunctionality for a small range of ρ values, specifically for $1.1 \lessapprox \rho \lessapprox 1.4$. The most accurate reconstruction of both orbits is achieved when $\rho = 1.25$. Much of the remaining results in this Chapter focus on exploring the dynamics of a multifunctional RC in this 'simplest hardest' case, 'simplest' in terms of the dynamics being limit cycles and 'hardest' as the training data is entirely overlapping.

Multifunctionality achieved

Fig. 6.6 provides examples of the short-term prediction of both x and y variables and trajectories on the reconstructed attractors, $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$, in comparison to $\mathcal{C}_A$ and $\mathcal{C}_B$ as viewed from $\mathbb{P}$ for $\rho = 1.1,\ 1.25,\ 1.4$. The direction of rotation on each cycle is indicated by the location of the large circular point and the nearby smaller circular point and the colours of these points are related to a given cycle. For instance, in the case of $\mathcal{C}_A$ which rotates anti-clockwise, the small circular point follows the larger circular point in an anti-clockwise direction. These figures highlight how values of δ_{rel}^{max} differ in regions where multifunctionality was achieved, for instance, $\delta_{rel}^{max} \approx 0.062, 0.019, 0.125$ for $\rho = 1.1,\ 1.25,\ 1.4$ respectively. These account for the slightly off-circular reconstructions of $\mathcal{C}_A$ for $\rho = 1.1$ and $\mathcal{C}_B$ for $\rho = 1.4$ as shown in Fig. 6.6.

Multifunctionality not achieved

Examples where the RC fails to achieve multifunctionality just outside the small window of ρ values identified in Fig. 6.4 are given in Fig. 6.7 for $\rho =$

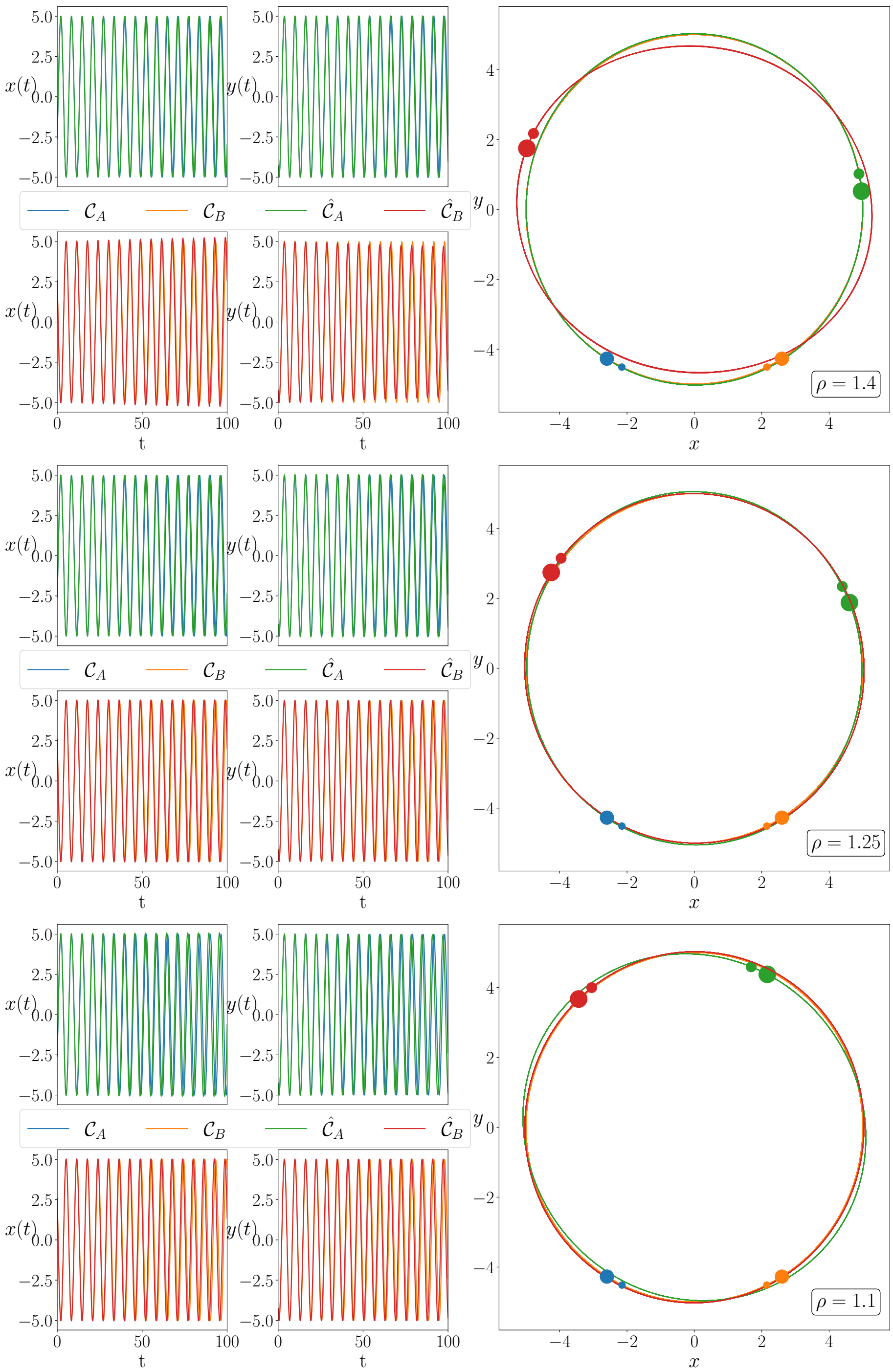


Figure 6.6: Examples where the RC solves the seeing double problem for $x_{cen} = 0$ at specified ρ values where t is the time after the prediction begins.

1.7 and 0.8. Here, the chaotic-like behaviour of the RC is seen for $\rho = 1.7$. What is important to note in this example is that shape swept out by the RC's trajectories when initialised from both $\hat{\boldsymbol{r}}(0) = \boldsymbol{r}_{\mathcal{C}_A}(t_{\text{listen}})$ and $\hat{\boldsymbol{r}}(0) = \boldsymbol{r}_{\mathcal{C}_B}(t_{\text{listen}})$ is bounded to regions nearby the original circular orbits, however the direction of rotation is not preserved as the RC switches between oscillating in clockwise and anti-clockwise directions.

For $\rho = 0.8$, Fig. 6.7 shows that when the reconstruction of both $\mathcal{C}_A$ and $\mathcal{C}_B$ fails, in correspondence with the result shown earlier in Fig 6.3, then the state of the RC tends towards the same limit cycle which bears no resemblance to either $\mathcal{C}_A$ and $\mathcal{C}_B$. The role of this limit cycle is assessed later in this Chapter.

6.4.3 Projected basins of attraction for $x_{cen} = 0$

The focus of this section is to provide some insight towards the nature of the basins of attraction created by the RC in order to achieve multifunctionality in the extreme case of completely overlapping orbits for $x_{cen} = 0$.

Method of generating the basins

Considering the dimension of the RC's state space, determining the precise location of the basins of attraction for both $\mathcal{A}$ and $\mathcal{B}$ is highly expensive. Instead, how the state of the RC evolves from its representation of different points in $\mathbb{R}^D$ is considered. From this it is possible to see how $\mathbb{P}$ is split to accommodate for the coexistence of $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ when these are completely overlapping.

To be more specific, after the RC has been trained for a given ρ and $x_{cen} = 0$, a point in $\mathbb{R}^D$ is chosen to drive the corresponding listening RC (Eq. (2.10)) with this point from $t = 0$ to $t = t_{\text{listen}}$. After this length of time, the state of the RC converges to a representation of the point from $\mathbb{R}^D$ to $\mathbb{R}^N$. This response vector is labelled as $\boldsymbol{r}_D = \boldsymbol{r}(t_{\text{listen}})$. The predicting RC is then initialised with $\hat{\boldsymbol{r}}(0) = \boldsymbol{r}_D$ and the long term behaviour of the prediction is characterised from this IC.

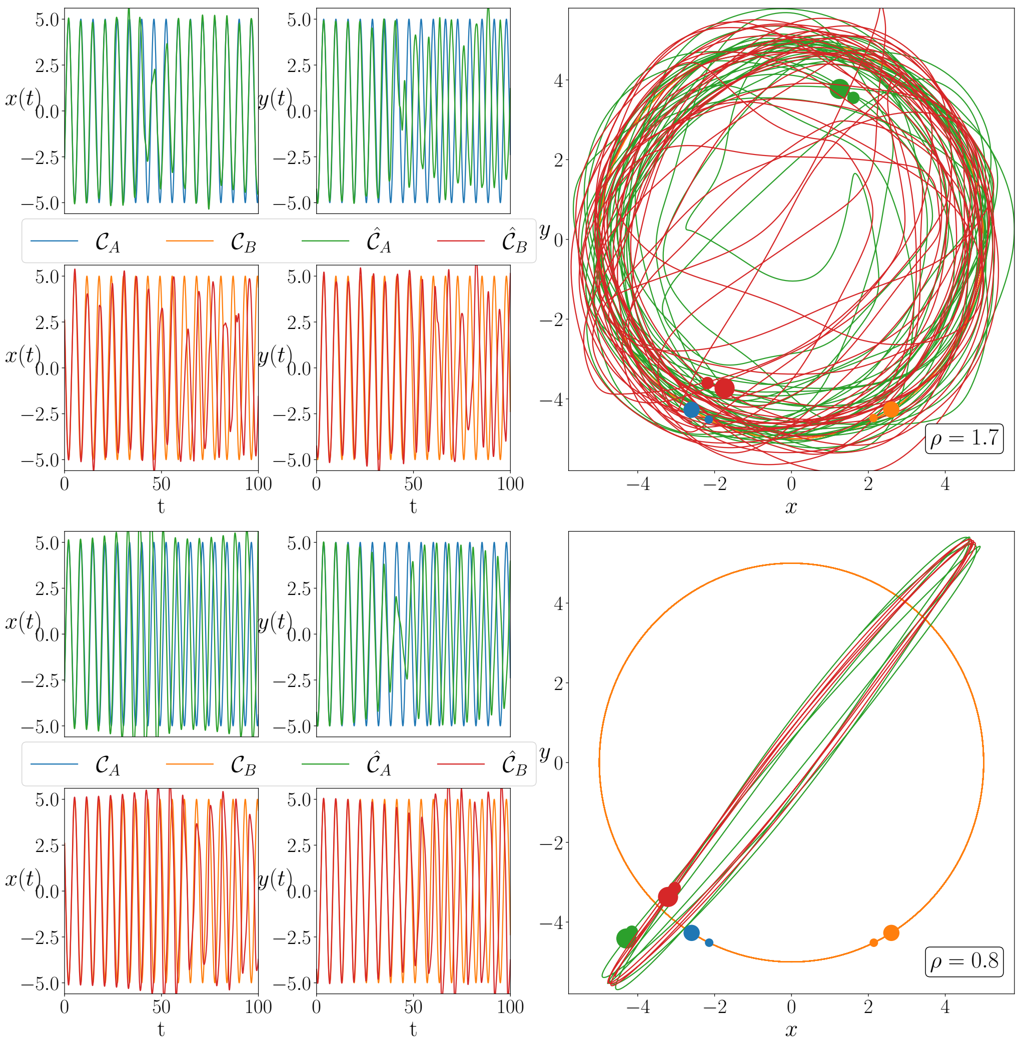


Figure 6.7: Examples where the RC fails to solve the seeing double problem for $x_{cen} = 0$ at specified ρ values where t is the time after the prediction begins.

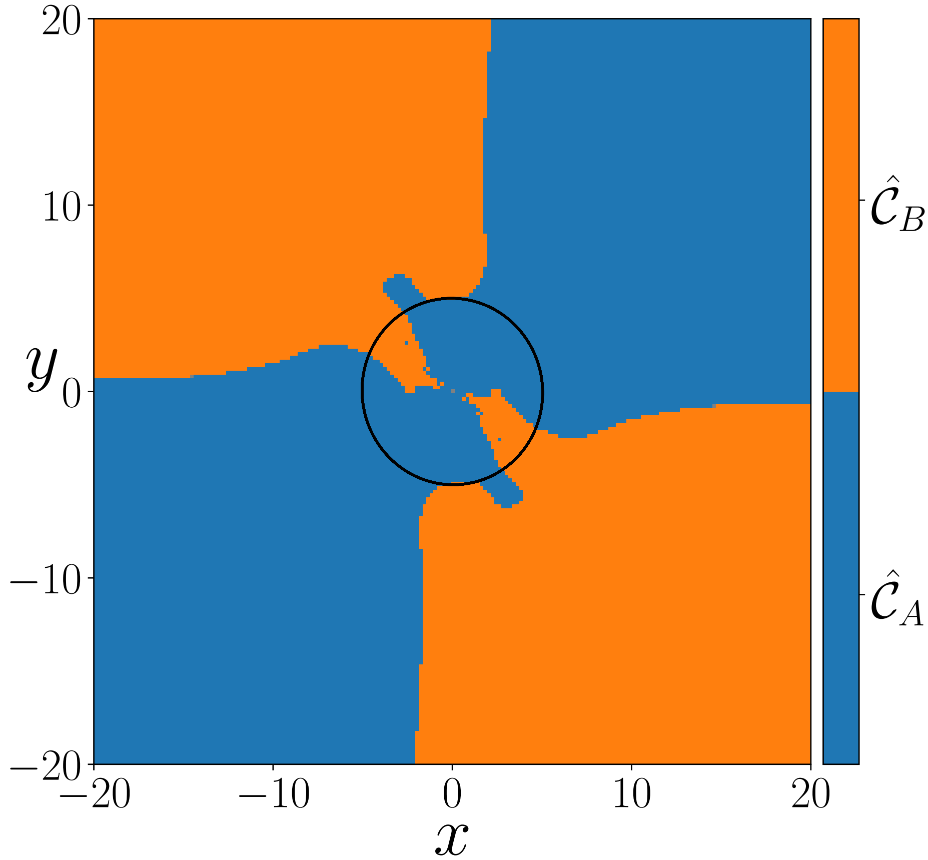


Figure 6.8: Basins of attraction for $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$, coloured in blue and orange respectively, as viewed from $\mathbb{P}$ for $\rho = 1.25$. Shown in black is an outline of the attractors location. In grey is the location of an unstable fixed point at the origin.

$\rho = 1.25$

Fig. 6.8 provides a picture of the projected basins of attraction in $\mathbb{P}$ using the above method when multifunctionality was achieved with the best accuracy for $\rho = 1.25$. From this it can be seen how $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ share $\mathbb{P}$. Each colour in this picture denotes which attractor the RC evolves towards when initialised from a given representation of a point in $\mathbb{R}^D$. If the state of the RC converges to $\hat{\mathcal{C}}_A$ then the point is coloured blue and similarly for $\hat{\mathcal{C}}_B$ the point is coloured orange. There is one exception, when the RC is driven with $(0, 0)$ the state remains at the fixed point at the origin, this is coloured grey.

The reconstructed attractors, $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$, are both plotted in black.

Fig. 6.8 shows that the number of points which converge to either attractor is not evenly distributed. In particular, most of the points closest to the location of the reconstructed attractors in $\mathbb{P}$ tend towards $\hat{\mathcal{C}}_A$. Furthermore, there is no single curve which entirely divides the boundary between the basins of $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ in Fig. 6.8, at the same time, a more regular or complex pattern may be occur in $\mathbb{S}$. Nonetheless, the method used to produce Fig. 6.8 provides an indication that for $\rho = 1.25$, the basin of attraction for the attractor $\mathcal{A}$ in $\mathbb{S}$, which when projected to $\mathbb{P}$ describes $\hat{\mathcal{C}}_A$, is of a larger volume than the basin for $\mathcal{B}$ in $\mathbb{S}$.

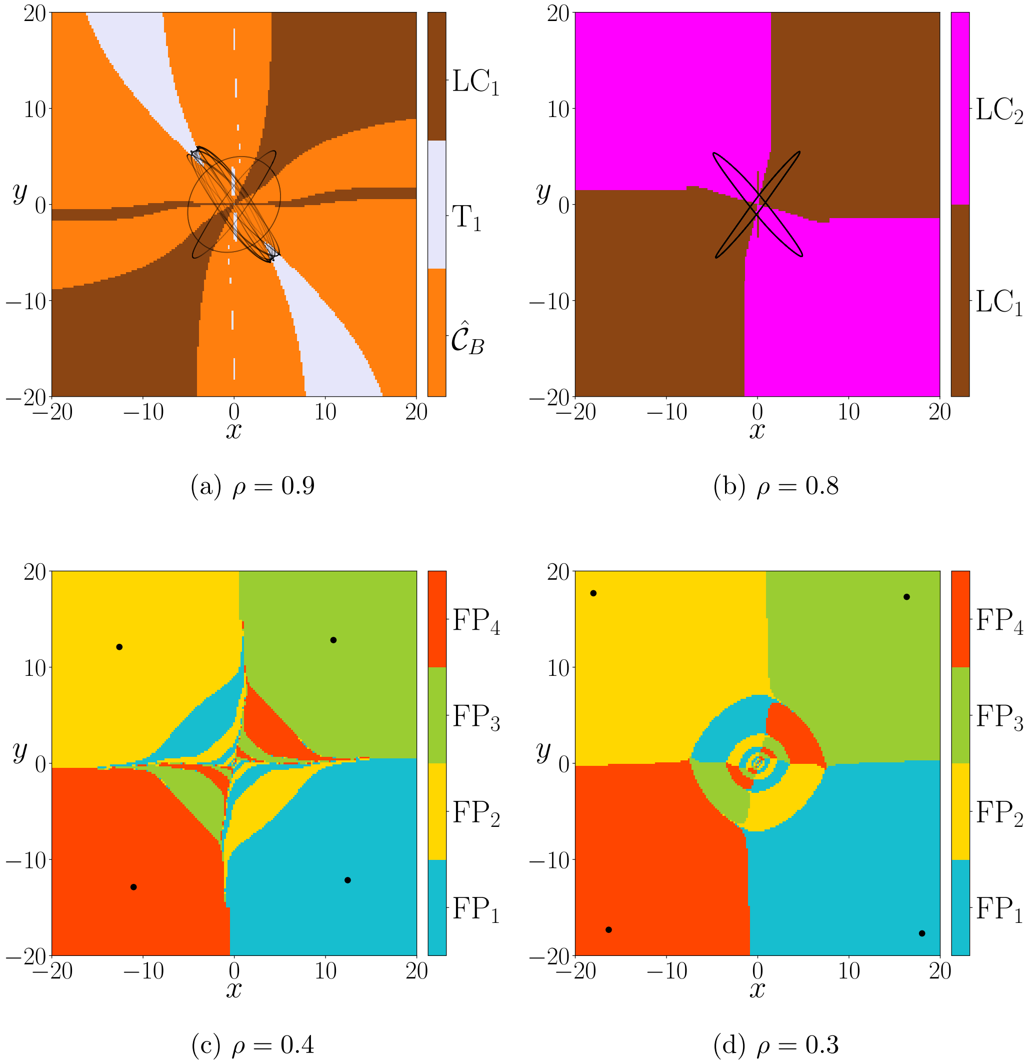
$\rho = 0.9, 0.8, 0.4, 0.3$

In order to gain a broader understanding of the RC's dynamics when it fails to distinguish between $\mathcal{C}_A$ and $\mathcal{C}_B$, the different basins of attraction which exist before the RC exhibits multifunctionality are now studied. The same procedure which generated the basins for $\rho = 1.25$ is now used to find the projected basins of attraction in $\mathbb{P}$ when setting $\rho = 0.9, 0.8, 0.4$ and, 0.3 in the case of $x_{cen} = 0$.

Prior to achieving multifunctionality, Fig. 6.3 shows that the RC is able to reconstruct $\mathcal{C}_B$ before $\mathcal{C}_A$. For $\rho = 0.9$, Fig. 6.9a illustrates that in addition to $\hat{\mathcal{C}}_B$ there are other basins of attraction for a limit cycle labelled as LC_1 and a quasi-periodic torus labelled as T_1. So while the aim is to train the RC to reconstruct both $\mathcal{C}_A$ and $\mathcal{C}_B$, there are other attractors which can be reached from an abundance of ICs.

In the case where neither $\mathcal{C}_A$ or $\mathcal{C}_B$ can be reconstructed, for example, when $\rho = 0.8$ as seen in Fig. 6.3 and Fig. 6.7, Fig. 6.9b shows that there are basins of attraction for two different limit cycles, LC_1 from Fig. 6.9a and another limit cycle denoted as LC_2. So at one point before achieving multifunctionality there is a coexistence of two different limit cycles in $\mathbb{P}$.

When setting $\rho = 0.4$, Fig. 6.9c reveals that four fixed points FP_1, FP_2, FP_3, and FP_4 are the only stable attractors which populate $\mathbb{P}$. Far away from the origin it can be seen that $\mathbb{P}$ is split into nearly equal quadrants.



(a) $\rho = 0.9$

(b) $\rho = 0.8$

(c) $\rho = 0.4$

(d) $\rho = 0.3$

Figure 6.9: Basins of attraction for $\hat{\mathcal{C}}_B$, LC_1, T_1, LC_2, FP_1, FP_2, FP_3, and FP_4, coloured in orange, brown, lavender, fuchsia, blue, yellow, green, and red respectively, as viewed from $\mathbb{P}$ when setting $x_{cen} = 0$ and (a) $\rho = 0.9$, (b) $\rho = 0.8$, (c) $\rho = 0.4$, and (d) $\rho = 0.3$. Shown in black is an outline of the attractors location in each figure. In grey is the location of an unstable fixed point at the origin.

However, closer to the origin there is a greater interaction between the basins resulting in the emergence of a fractal pattern. A similar result is found when setting $\rho = 0.3$ in Fig. 6.9d with fractal basins boundaries appearing closer to the origin. For further reading on fractal basin boundaries and the difficulties these cause in predicting the long-term behaviour of a dynamical system see [146–148].

6.4.4 Symmetry and Seeing Double

It is important to note that in Figs. 6.8 and 6.9 there are certain symmetries present in the different projected basins of attraction.

In Figs. 6.8, 6.9a, and 6.9b, the same attractor is reached whether the RC is initialised with its representation of either the point (x', y') or $(-x', -y')$. However, there is a slightly different situation in Figs. 6.9c-6.9d. Using the terminology developed by Herteux and Räth [77] and used previously in Chapter 5, when the RC is initialised with its representation of the point (x', y') the state settles down to a particular fixed point, for example FP_1 or FP_3, if instead the RC was initialised with its representation of $(-x', -y')$ then the state of the RC settles on the 'mirror-attractor' of that fixed point, in this case FP_2 or FP_4.

Despite breaking the symmetry in this RC setup, by including the square terms in Eq. (2.13) in the RC design, there are certain properties of the seeing double training data which allow for the emergence of mirror-attractors in a similar way to some of the results presented in Chapter 5 regarding the 'symmetry kills the square' phenomenon. The manner in which these scenarios arise in the case of the seeing double problem are outlined in the remainder of this section.

Seeing Double and $\mathbf{W}^{(2)}_{out} \to \mathbf{0}$

In order for mirror-attractors to come into existence, the symmetry breaking terms in the RC setup studied here, i.e., the $\boldsymbol{r}^2(t)$ term in $\boldsymbol{q}(\boldsymbol{r}(t)) = (\boldsymbol{r}(t)\ \boldsymbol{r}^2(t))^T$ and the corresponding $\mathbf{W}^{(2)}_{out}$ term in $\mathbf{W}_{out} = \left(\mathbf{W}^{(1)}_{out}\ \mathbf{W}^{(2)}_{out}\right)$, cannot exist.

In the case of the seeing double problem, there are two conditions which contribute to forcing $\mathbf{W}_{out}^{(2)} \to \mathbf{0}$ that need to be satisfied simultaneously. These are the anti-symmetric nature of the individual circular orbits, as described by Eq. (5.17), and the overall anti-symmetric nature of the combined training data set described in Eqs. (5.5) and (5.6) which determine $\mathbf{W}_{out}$.

The first of the conditions identified above, which relates to Eq. (5.17), holds when the RC is trained to reconstruct only $\mathcal{C}_A$ or $\mathcal{C}_B$ as,

$$\boldsymbol{u}_{\mathcal{C}_A}(t) = -\boldsymbol{u}_{\mathcal{C}_A}(t + \frac{T}{2}) \quad \text{and} \quad \boldsymbol{u}_{\mathcal{C}_B}(t) = -\boldsymbol{u}_{\mathcal{C}_B}(t + \frac{T}{2}), \tag{6.3}$$

is satisfied $\forall\, t$ so long as the RC responds to this driving input such that,

$$\boldsymbol{r}_{\mathcal{C}_A}(t) = -\boldsymbol{r}_{\mathcal{C}_A}(t + \frac{T}{2}) \quad \text{and} \quad \boldsymbol{r}_{\mathcal{C}_B}(t) = -\boldsymbol{r}_{\mathcal{C}_B}(t + \frac{T}{2}), \tag{6.4}$$

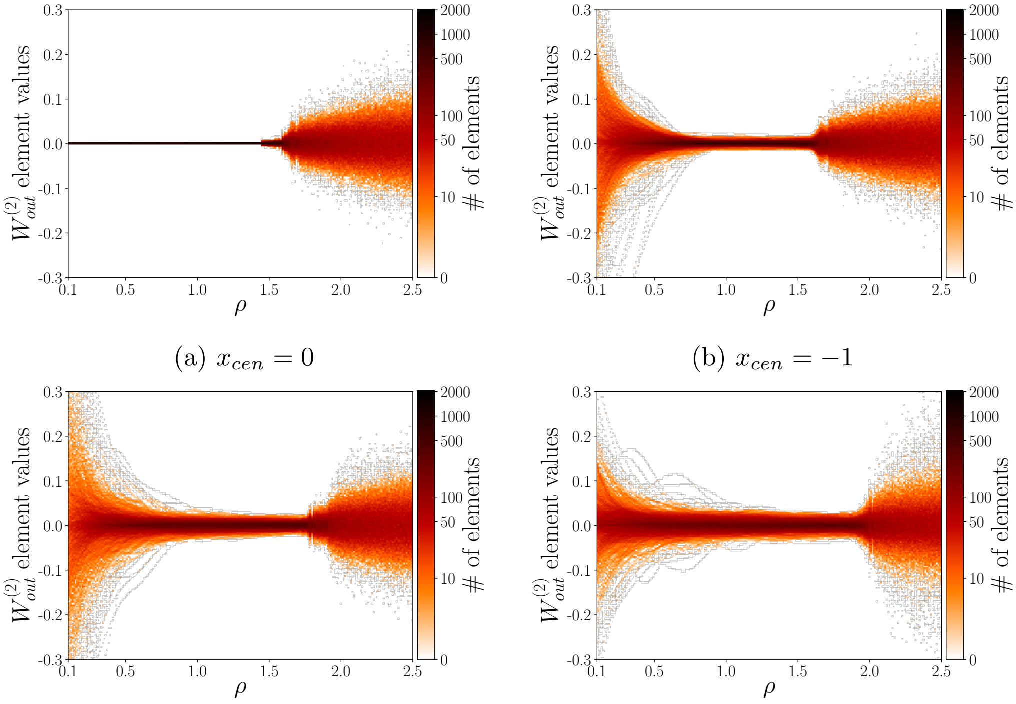
where T is the period of oscillation on each circular orbit.

However, complications arise with the second condition which concerns the structure of the combined training data sets, Eqs. (5.5) and (5.6). On one hand, this condition holds $\forall\, x_{cen}$ when $\mathcal{C}_A$ and $\mathcal{C}_B$ are rotating in the same direction as the RC is effectively being trained to reconstruct mirror-attractors. To be more specific, the input data, $\boldsymbol{u}(t)$, from, for instance, $\mathcal{C}_A$ when centered at $(x_{cen}, 0)$, is equal to $-\boldsymbol{u}(t)$ when $\mathcal{C}_A$ is centered at $(-x_{cen}, 0)$, denoted as $\mathcal{C}_A^-$. As the RC is trained to reconstruct a coexistence of $\mathcal{C}_A$ and $\mathcal{C}_B$ for a given x_{cen}, and $\mathcal{C}_B = \mathcal{C}_A^- = -\mathcal{C}_A$ then $\boldsymbol{u}_{\mathcal{C}_B}(t) = -\boldsymbol{u}_{\mathcal{C}_A}(t)$ and

$$\begin{pmatrix} \boldsymbol{r}_{\mathcal{C}_B}(t) \\ \boldsymbol{r}^2_{\mathcal{C}_B}(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{r}_{\mathcal{C}_A^-}(t) \\ \boldsymbol{r}^2_{\mathcal{C}_A^-}(t) \end{pmatrix} = \begin{pmatrix} -\boldsymbol{r}_{\mathcal{C}_A}(t) \\ (-\boldsymbol{r}_{\mathcal{C}_A}(t))^2 \end{pmatrix} = \begin{pmatrix} -\boldsymbol{r}_{\mathcal{C}_A}(t) \\ \boldsymbol{r}^2_{\mathcal{C}_A}(t) \end{pmatrix}. \tag{6.5}$$

Therefore, the resulting $\mathbf{X}$ and $\mathbf{Y}$ training data matrices are of the form described in Eqs. (5.5) and (5.6) and as a consequence, $\mathbf{W}_{out}^{(2)} = \mathbf{0}$.

On the other hand, when $\mathcal{C}_A$ and $\mathcal{C}_B$ are rotating in opposite directions there is no mirror symmetry present in the combined training data sets. As a result, it is only when $x_{cen} = 0$ that 'symmetry kills the square' as the combined training data matrices from both $\mathcal{C}_A$ and $\mathcal{C}_B$ satisfy Eq. (5.17). As a further comment, if instead the RC was trained to reconstruct a coexistence of $\mathcal{C}_A$ and $\mathcal{C}_B$ when these are of different radii, so long as Eq. (5.17) is satisfied



(a) $x_{cen} = 0$ (b) $x_{cen} = -1$

(c) $x_{cen} = -5$ (d) $x_{cen} = -10$

Figure 6.10: Histograms of the $\mathbf{W}_{out}^{(2)}$ elements for $\rho \in [0.1, 2.5]$ when in (a) $x_{cen} = 0$, in (b) $x_{cen} = -1$, in (c) $x_{cen} = -5$, and in (d) $x_{cen} = -10$.

for both $\boldsymbol{u}_{\mathcal{C}_A}(t)$ and $\boldsymbol{u}_{\mathcal{C}_B}(t)$ then $\mathbf{W}_{out}^{(2)} = \mathbf{0}$ as the RC is trained to provide a coexistence of attractors which are all anti-symmetric to themselves. In return, this places no constraint on the behaviour of the corresponding response attractors, $\mathcal{A}$ and $\mathcal{B}$, in $\mathbb{S}$ as these do not necessarily need to be mirror-attractors of one another as already shown in Fig. 6.8.

In addition, the above discussion provides further insight towards how multifunctionality is achieved in the first place as, even though the only difference between the training data from both $\mathcal{C}_A$ and $\mathcal{C}_B$ is the direction of rotation, the RC is required to respond to the driving input data in such a way that the corresponding input driven attractors, $\mathcal{A}$ and $\mathcal{B}$, can coexist in $\mathbb{S}$ and resemble $\mathcal{C}_A$ and $\mathcal{C}_B$ when projected to $\mathbb{P}$.

$\boldsymbol{\rho}$ vs. $\mathbf{W}_{\boldsymbol{out}}^{\mathbf{(2)}}$

Given the insight gained already from Fig. 5.9, and the role of ρ in the behaviour of $\mathbf{W}_{out}^{(2)}$, the same procedure is applied here for the present case of the seeing double problem ($\mathcal{C}_A$ and $\mathcal{C}_B$ rotating in opposite directions) when $x_{cen} = 0$ and the result of this is illustrated in Fig. 6.10a. There is a threshold value of ρ where $\mathbf{W}_{out}^{(2)}$ comes into existence. However, since $\mathbf{W}_{out}^{(2)} = \mathbf{0}$ for all ρ values less than this critical value, each solution of the predicting RC is required to either have a mirror-attractor or is anti-symmetric to itself, whether or not multifunctionality is achieved.

Nevertheless, the existence of mirror-attractors in Figs. 6.9c and 6.9d, which led towards this discussion in the first place, and that $\mathbf{W}_{out}^{(2)} = \mathbf{0}$ for the corresponding ρ values shown in Fig. 6.10a, together show that while both driving inputs possess the symmetry described in Eq. (6.3) and the RC may respond to these driving inputs according to Eq. 6.4, then it is still possible for $\mathbf{W}_{out}^{(2)} = \mathbf{0}$ even if the resulting attractors that the RC reconstructs are not periodic but are instead mirrored fixed points.

When $x_{cen} \neq 0$ the symmetry in the training data is broken, which is much like shifting the location of the mirror-attractors in Sec. 5.3. The effect this has on the elements of $\mathbf{W}_{out}^{(2)}$ for $\rho \in [0.1, 2.5]$ is shown in Figs. 6.10b-6.10d. Similar to the results depicted in Fig. 5.3, the central message of Figs. 6.10b-6.10d is that as the magnitude of x_{cen} increases, from 1 to 10, so too does the magnitude of many elements of $\mathbf{W}_{out}^{(2)}$.

$$\left(\mathbf{W}_{\boldsymbol{out}}^{\mathbf{(1)}}\right)_{\boldsymbol{x_{cen}}} = \left(\mathbf{W}_{\boldsymbol{out}}^{\mathbf{(1)}}\right)_{\boldsymbol{-x_{cen}}} \textbf{ and } \left(\mathbf{W}_{\boldsymbol{out}}^{\mathbf{(2)}}\right)_{\boldsymbol{x_{cen}}} = -\left(\mathbf{W}_{\boldsymbol{out}}^{\mathbf{(2)}}\right)_{\boldsymbol{-x_{cen}}}$$

While it is shown in Fig. 6.10a that $\mathbf{W}_{out}^{(2)} = \mathbf{0}$ for $x_{cen} = 0$, there is another $\mathbf{W}_{out}$ related phenomenon which arises from the 'symmetry kills the square' and mirror-attractor properties of the RC and structure of the seeing double training data that is produced as follows.

Building on the statements made earlier which led towards the relationship established in Eq. (6.5), $\boldsymbol{u}_{\mathcal{C}_A}(t) = -\boldsymbol{u}_{\mathcal{C}_A^-}(t)$ and the resulting input training data matrix, $\mathbf{Y}$, for a given x_{cen} and $-x_{cen}$ are related in the following way,

$$\mathbf{Y}_{x_{cen}} = -\mathbf{Y}_{-x_{cen}}. \tag{6.6}$$

Therefore, in order for the training to be successful and $\hat{\mathcal{C}_A} = -\hat{\mathcal{C}_A}^{-}$, where $\hat{\mathcal{C}_A}$ is the reconstructed version of $\mathcal{C}_A$ centered at $(x_{cen}, 0)$, then the following is also required,

$$(\mathbf{W}_{out})_{x_{cen}} \begin{pmatrix} \boldsymbol{r}_{\mathcal{C}_A}(t) \\ \boldsymbol{r}^2_{\mathcal{C}_A}(t) \end{pmatrix} = -\left(\mathbf{W}_{out}\right)_{-x_{cen}} \begin{pmatrix} \boldsymbol{r}_{\mathcal{C}_A^-}(t) \\ \boldsymbol{r}^2_{\mathcal{C}_A^-}(t) \end{pmatrix}, \tag{6.7}$$

where $(\mathbf{W}_{out})_{x_{cen}}$ is the resulting $\mathbf{W}_{out}$ matrix trained using data from $\mathcal{C}_A$ and $\mathcal{C}_B$ when centered at $(x_{cen}, 0)$ and $(-x_{cen}, 0)$ respectively for a given x_{cen}, and similarly for $(\mathbf{W}_{out})_{-x_{cen}}$ for the corresponding $-x_{cen}$.

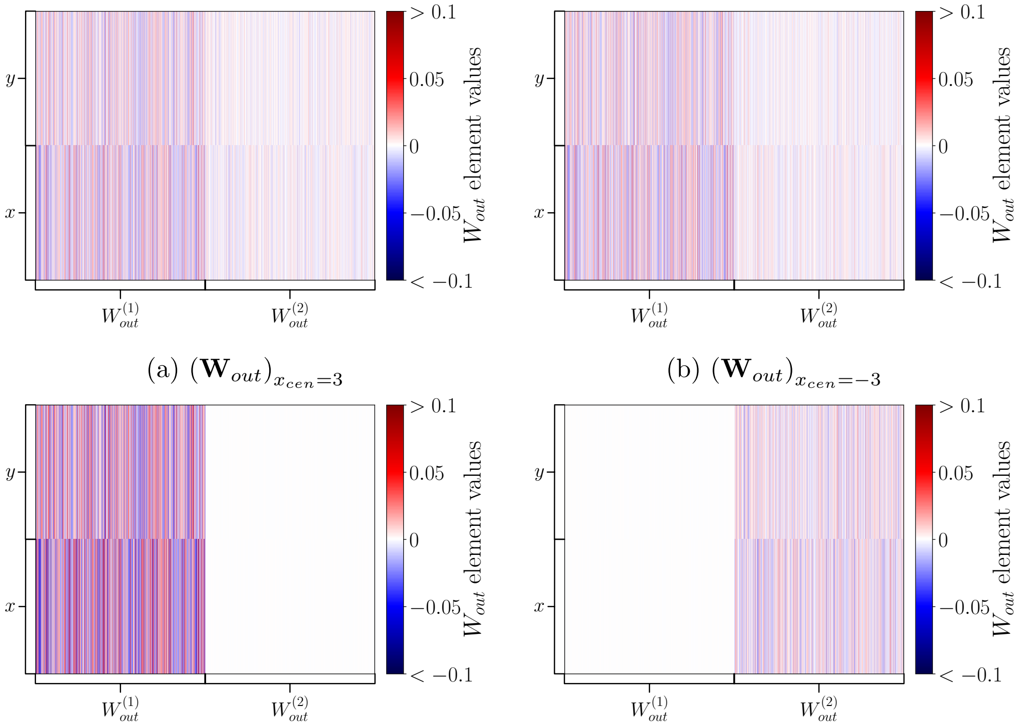
Using the information from Eq. (6.5), this allows Eq. (6.7) to be rewritten as,

$$(\mathbf{W}_{out})_{x_{cen}} \begin{pmatrix} -\boldsymbol{r}_{\mathcal{C}_A^-}(t) \\ \boldsymbol{r}^2_{\mathcal{C}_A^-}(t) \end{pmatrix} = -\left(\mathbf{W}_{out}\right)_{-x_{cen}} \begin{pmatrix} \boldsymbol{r}_{\mathcal{C}_A^-}(t) \\ \boldsymbol{r}^2_{\mathcal{C}_A^-}(t) \end{pmatrix} \tag{6.8}$$

from which it follows that,

$$\begin{aligned}
&-\left(\mathbf{W}^{(1)}_{out}\right)_{x_{cen}} \boldsymbol{r}_{\mathcal{C}_A^-}(t) + \left(\mathbf{W}^{(2)}_{out}\right)_{x_{cen}} \boldsymbol{r}^2_{\mathcal{C}_A^-}(t) \\
&= -\left(\mathbf{W}^{(1)}_{out}\right)_{-x_{cen}} \boldsymbol{r}_{\mathcal{C}_A^-}(t) - \left(\mathbf{W}^{(2)}_{out}\right)_{-x_{cen}} \boldsymbol{r}^2_{\mathcal{C}_A^-}(t), && (6.9) \\
&\implies \left(\left(\mathbf{W}^{(2)}_{out}\right)_{x_{cen}} + \left(\mathbf{W}^{(2)}_{out}\right)_{-x_{cen}}\right) \boldsymbol{r}^2_{\mathcal{C}_A^-}(t) \\
&= \left(\left(\mathbf{W}^{(1)}_{out}\right)_{x_{cen}} - \left(\mathbf{W}^{(1)}_{out}\right)_{-x_{cen}}\right) \boldsymbol{r}_{\mathcal{C}_A^-}(t), && (6.10) \\
&\therefore \left(\mathbf{W}^{(1)}_{out}\right)_{x_{cen}} = \left(\mathbf{W}^{(1)}_{out}\right)_{-x_{cen}}, \text{ and } \left(\mathbf{W}^{(2)}_{out}\right)_{x_{cen}} = -\left(\mathbf{W}^{(2)}_{out}\right)_{-x_{cen}}. && (6.11)
\end{aligned}$$

The above analytical results in Eq. (6.11) are confirmed numerically in Fig. 6.11 by examining the behaviour of the resulting $(\mathbf{W}_{out})_{x_{cen}}$ matrices after training the RC with data from $\mathcal{C}_A$ and $\mathcal{C}_B$ for $x_{cen} = 3$ in Fig. 6.11a and $x_{cen} = -3$ in Fig. 6.11b where $\rho = 1.2$ in both cases. The elements of the resulting matrix from the element-wise addition of $(\mathbf{W}_{out})_{x_{cen}=3}$ and $(\mathbf{W}_{out})_{x_{cen}=-3}$



(a) $(\mathbf{W}_{out})_{x_{cen}=3}$

(b) $(\mathbf{W}_{out})_{x_{cen}=-3}$

(c) $(\mathbf{W}_{out})_{x_{cen}=3} + (\mathbf{W}_{out})_{x_{cen}=-3}$

(d) $(\mathbf{W}_{out})_{x_{cen}=3} - (\mathbf{W}_{out})_{x_{cen}=-3}$

Figure 6.11: $(\mathbf{W}_{out})_{x_{cen}}$ elements for $x_{cen} = 3$ in (a) and $x_{cen} = -3$ in (b). Result of element-wise addition of $(\mathbf{W}_{out})_{x_{cen}=3} + (\mathbf{W}_{out})_{x_{cen}=-3}$ in (c) and element-wise subtraction of $(\mathbf{W}_{out})_{x_{cen}=3} - (\mathbf{W}_{out})_{x_{cen}=-3}$ in (d). Same convention of plotting the elements of $(\mathbf{W}_{out})_{x_{cen}}$ as described in Fig. 5.2 is used here.

is depicted in Fig. 6.11c which shows that $\left(\mathbf{W}^{(2)}_{out}\right)_{x_{cen}} + \left(\mathbf{W}^{(2)}_{out}\right)_{-x_{cen}} = \mathbf{0}$. Similarly, in Fig. 6.11d, the element-wise subtraction of $(\mathbf{W}_{out})_{x_{cen}=-3}$ from $(\mathbf{W}_{out})_{x_{cen}=3}$ results in $\left(\mathbf{W}^{(1)}_{out}\right)_{x_{cen}} - \left(\mathbf{W}^{(1)}_{out}\right)_{-x_{cen}} = \mathbf{0}$.

While these results are interesting from an analytical and numerical perspective, their influence has been a common factor throughout many of the results presented in this Chapter. For instance, the symmetries seen in Figs. 6.3-6.5 arise due to the constraint described in Eq. (6.11) as imposed by the structure of the training data. These results regarding symmetry and the seeing double problem continue to play a significant role in the following section.

6.5 Seeing Double Dynamics

The results presented in Sec. 6.4 outline where multifunctionality is achieved in the (x_{cen}, ρ)-plane by assessing the long term dynamics of the trained RC. However, from Figs. 6.3a, 6.3b, and 6.9, it appears that the RC must overcome the influence of several other attractors before either $\mathcal{C}_A$ or $\mathcal{C}_B$ can be reconstructed. It is in this section that the interplay between these attractors and the reconstructed attractors is investigated and the particular bifurcations through which the RC learns how to solve the seeing double problem are identified.

6.5.1 Bifurcation Analysis for $x_{cen} = 0$

In Sec. 6.4.3 multiple fixed points, limit cycles, and a torus were found to coexist in $\mathbb{P}$ at different ρ values. The symmetrical properties of these attractors were analysed in Sec. 6.4.4. However, according to the results shown in Fig. 6.8, these attractors seemingly vanish from $\mathbb{P}$ as multifunctionality is achieved. Moreover, for a small change in ρ there is also a small change in the location of these fixed points and the nature of these limit cycles. This intriguing behaviour warrants further investigation in order to establish the suggested link between these attractors and the reconstructed attractors.

Taking the above statements into account, the evolution of the fixed points, FP_1, FP_2, FP_3, and FP_4, shown in Figs. 6.9c-6.9d are now tracked with respect to ρ using the same procedure described at the beginning of Sec. 4.5. The results of this as viewed from $\mathbb{P}$ are shown in Fig. 6.12.

Overall, Fig. 6.12 provides a clear picture on the challenges which the RC must overcome in order to achieve multifunctionality. One of the most distinctive features of Fig. 6.12 is the indication that, as ρ is increased the dynamics of the RC begins to bear a stronger resemblance to $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$.

For small ρ, Fig. 6.12 shows that the RC is unable to create any time-varying dynamics and instead produces two pairs of antisymmetric fixed points. As ρ is increased, Fig. 6.12 establishes that these branches of mirrored fixed points are drawn closer together and two limit cycles are born in

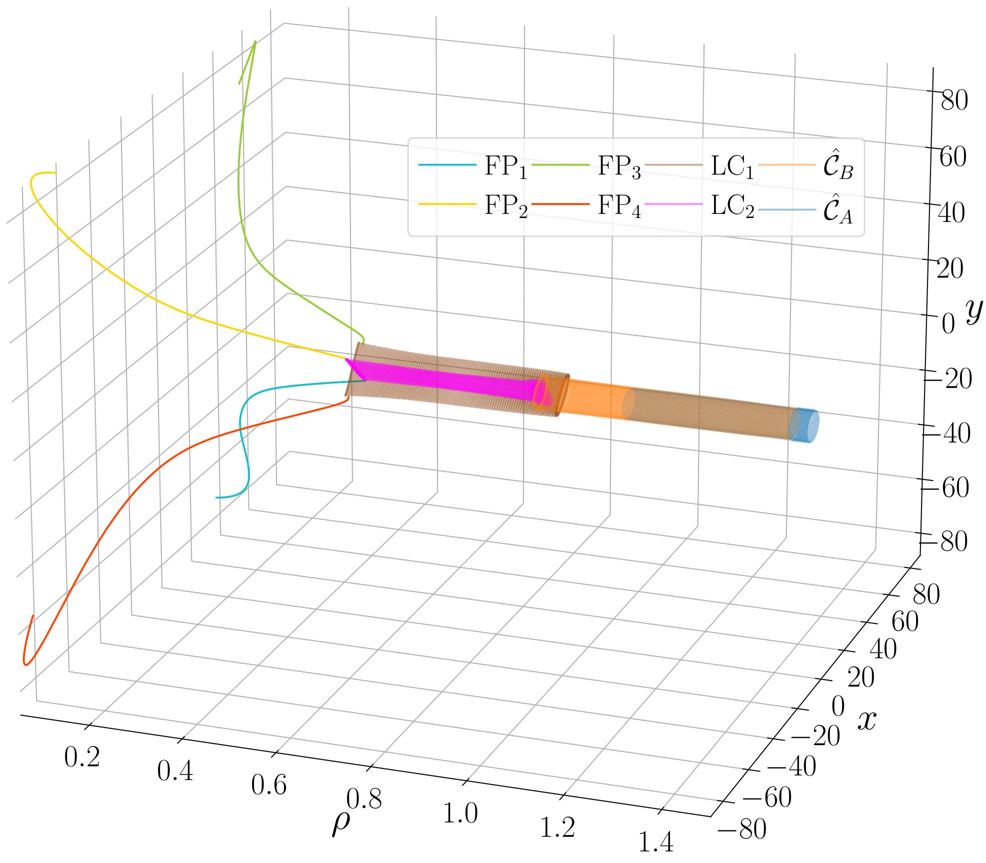


Figure 6.12: Bifurcation diagram of stable attractors in $\mathbb{P}$ for $\rho \in [0.01, 1.47]$.

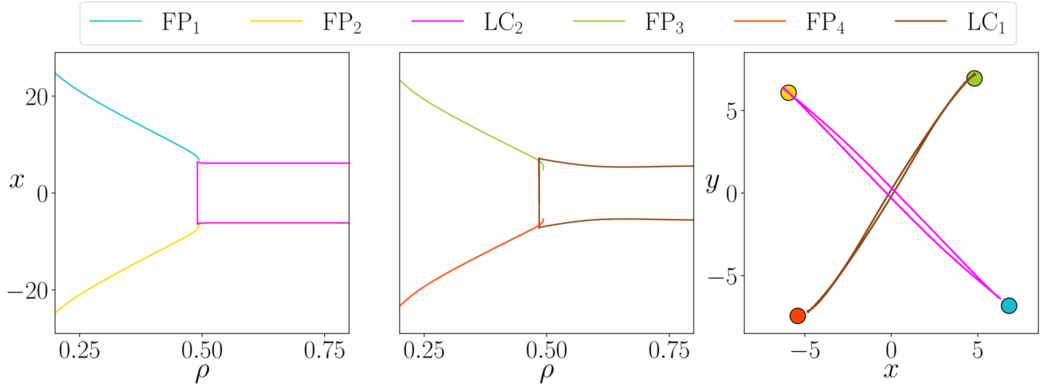


Figure 6.13: Left and middle panels illustrate the bifurcations from FP$_1$ and FP$_2$ to LC$_1$, and from FP$_3$ and FP$_4$ to LC$_2$, for $\rho \in [0.2, 0.8]$ in the (ρ, x)-plane. Right panel shows the location of the fixed points and limit cycles in $\mathbb{P}$ just before these attractors can no longer be tracked.

quick succession of one another. These are the same limit cycles, LC$_1$ and LC$_2$, that were found previously in Fig. 6.9b. The role played by these fixed points and limit cycles is akin to the formation of a memory in the brain. As the RC learns the correct 'memory', in this case the correct attractor, along the way its dynamics becomes more similar to the attractor as the influence of the training data grows until attractor reconstruction is achieved.

Once ρ is sufficiently large, Fig. 6.12 shows that the RC is first able to reconstruct $\mathcal{C}_B$ and subsequently for larger ρ, the RC achieves multifunctionality by reconstructing a coexistence of $\mathcal{C}_A$ and $\mathcal{C}_B$.

A more comprehensive breakdown on each of the major events described above is provided in the remainder of the present subsection.

Bifurcations of FP$_1$ and FP$_2$ to LC$_1$, and of FP$_3$ and FP$_4$ to LC$_2$

Fig. 6.12 shows that as ρ is increased from 0.01, the branches of mirror fixed points, FP$_1$ and FP$_2$, FP$_3$ and FP$_4$, behave according to the dynamical restraints imposed by the 'symmetry kills the square' phenomenon as $\mathbf{W}_{out}^{(2)} = \mathbf{0}$ for $\rho < 1.44$ as shown in Fig. 6.10a.

Fig. 6.13 provides a more detailed picture on the bifurcations from FP$_1$ and FP$_2$ to LC$_2$ at $\rho = 0.4935$ in the left hand panel and from FP$_3$ and FP$_4$

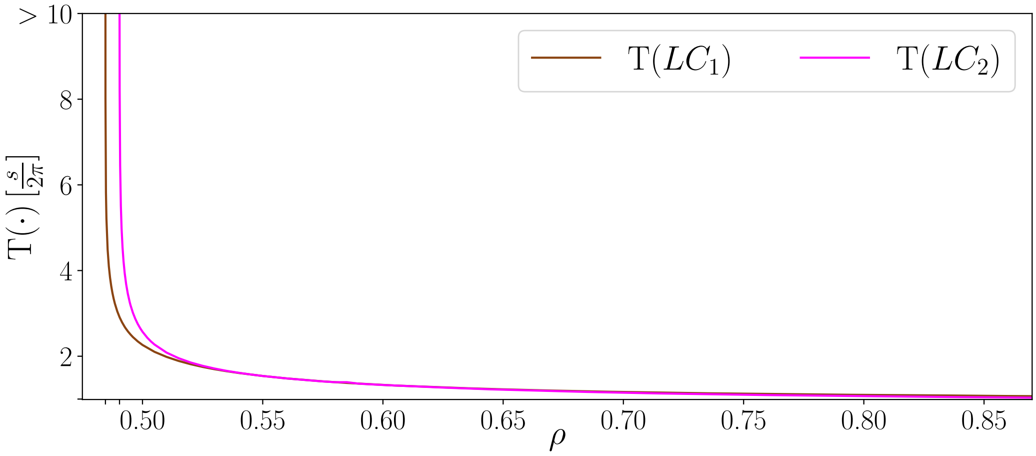


Figure 6.14: Period, T($\cdot$), of LC_1 (in brown) and LC_2 (in fuchsia) in units of seconds/2π versus ρ.

to LC_1 at $\rho = 0.4931$ in the middle panel. Plotted here are the location of the fixed points and the local maxima and minima of the limit cycles in the (ρ, x)-plane. When tracking the evolution of these limit cycles for decreasing ρ, it is found that LC_2 can no longer be tracked for $\rho < 0.4905$ and similarly for LC_1 for $\rho < 0.4847$

The panel on the right of Fig. 6.13 displays the location of the fixed points and limit cycles just before and after the respective bifurcations take place on each fixed point and limit cycle. It also appears here that the location of the fixed points just prior to the bifurcation are related to the amplitude of oscillation for both limits cycles after the bifurcation.

In Fig 6.14, it is shown that as ρ approaches the point at which LC_1 and LC_2 can no longer be tracked, at $\rho \approx 0.4847$ and 0.4905 respectively, the period, T($\cdot$), of both limit cycles tends to infinity. T($\cdot$) is plotted here in units of seconds/2π to provide a comparison to the period of the driving input signals from $\mathcal{C}_A$ and $\mathcal{C}_B$ which correspond to $\mathrm{T}(\cdot) = 1$.

Fig 6.14 also shows that as ρ increases, $\mathrm{T}(LC_1)$ and $\mathrm{T}(LC_2)$ converge to 1. This provides further indication that while there does not appear to be a smooth transition from either LC_1 or LC_2 to $\hat{\mathcal{C}}_A$ or $\hat{\mathcal{C}}_B$, these 'untrained dynamics' bear a stronger resemblance to the desired dynamics as ρ increases.

Taking all of the above information on LC_1 and LC_2 into account, there are two potential bifurcations which describe how these transitions from two fixed points to one limit cycle takes place. On one hand, since there appears to be a small range of ρ values where FP_1, FP_2, and LC_2 coexist for $\rho \in (0.4905, 0.4935)$ and similarly where FP_3, FP_4, and LC_1 coexist for $\rho \in (0.4847, 0.4931)$, then a heteroclinic bifurcation may be responsible for the birth of these stable limit cycles. On the other hand, since these windows of multistability appear for only a small range of ρ values, their existence may be an artefact of the tracking procedure used to compute the evolution of these fixed points and limit cycles with regard to changes in ρ. If there is no window of multistability then a saddle-node infinite period (SNIPER) bifurcation may instead be responsible for the transition from antisymmetric pairs of fixed points to the corresponding limit cycle.

The above quandary identifies a pitfall of this tracking procedure as it is unable to detect precisely when, for instance, an attractor becomes unstable. Instead this approach relies on a suitably chosen step size of the bifurcation parameter, in this case ρ, to account for when a bifurcation takes place. If the step size is sufficiently small then this method of tracking the evolution of attractors can, for example, continue to track unstable solutions of the given dynamical system for a certain range of the bifurcation parameter values. There are of course other potential sources of error involved in these numerics which may effect how these windows of multistability occur close to a bifurcation. Despite these issues, the results presented in this section provide significant insight towards how the RC solves the seeing double problem and achieves multifunctionality.

Bifurcations of $\mathbf{LC_1}$ and $\mathbf{LC_2}$ and the birth of $\hat{\mathcal{C}}_B$

Fig. 6.12 also illustrates how the limit cycles, LC_1 and LC_2, evolve in $\mathbb{P}$ with regard to changes in ρ. An extensive account of this behaviour and the events which result in the death of these stable limit cycles and the creation of the first reconstructed cycle, $\hat{\mathcal{C}}_B$, is provided in Fig. 6.15.

The evolution of the local maxima of the RC's stable solutions in the

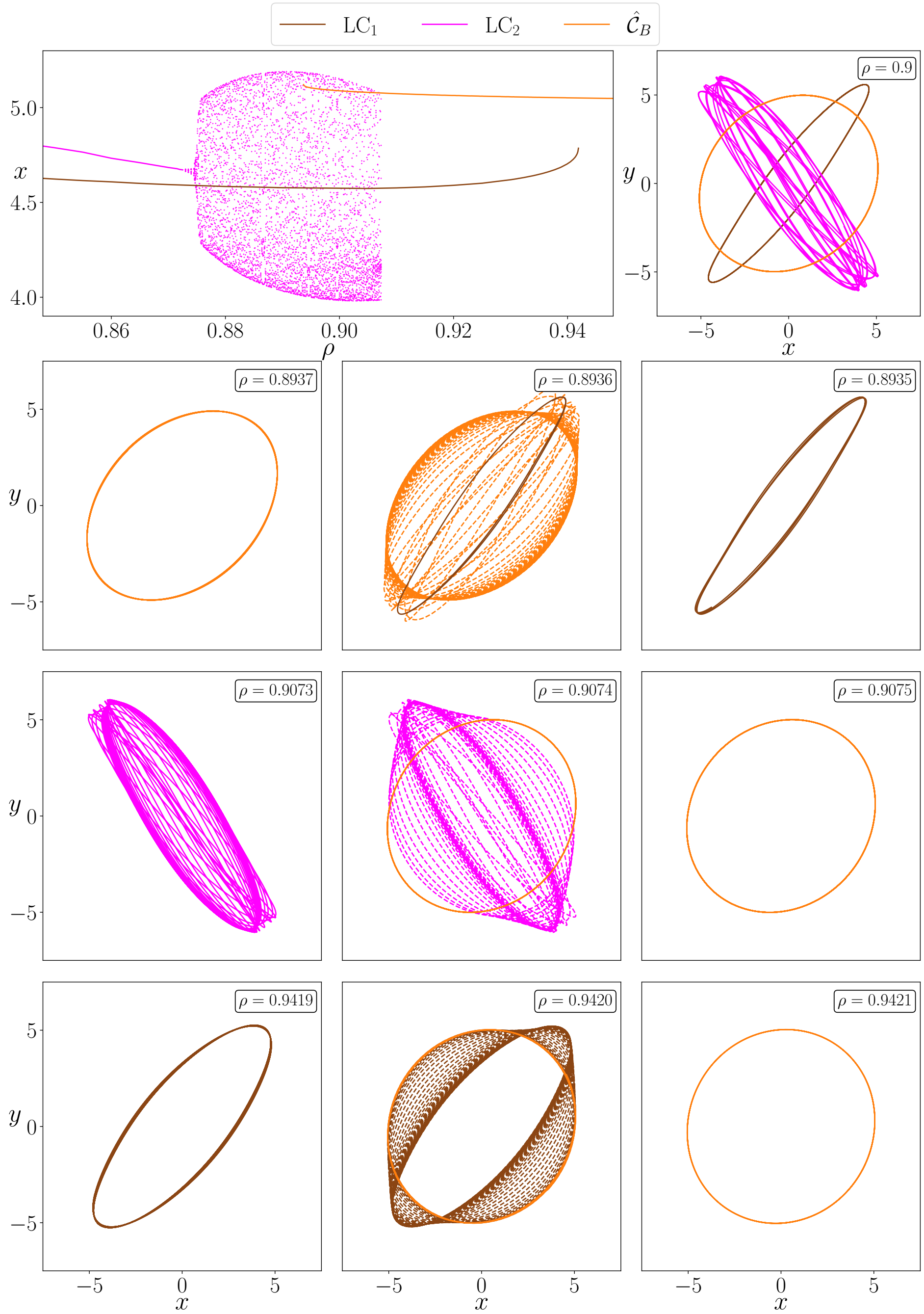


Figure 6.15: Top left panel illustrates how the local maxima of LC$_1$, LC$_2$, and $\hat{\mathcal{C}}_B$ change in the (ρ, x)-plane. Top right panel shows the attractors present in $\mathbb{P}$ for $\rho = 0.9$. The panels in the second, third, and fourth rows depict the dynamics of $\hat{\mathcal{C}}_B$, LC$_2$, and LC$_1$ respectively just before these attractors can no longer be tracked.

projected x coordinate for $\rho \in [0.848, 0.948]$ is plotted in the top left panel of Fig. 6.15. Here it can be seen that the quasi-periodic torus, T_1, shown earlier in Fig. 6.9a comes into existence due to LC_2 undergoing a torus bifurcation at $\rho = 0.8746$. In light of this, the top right panel provides an updated picture on the stable attractors which, in Fig. 6.9a, were shown to coexist in $\mathbb{P}$ for $\rho = 0.9$. The colour scheme in the top right panel is consistent with the bifurcation diagram shown in the top left panel instead of assigning the lavender colour to the torus as in Fig. 6.9a.

The second row of images in Fig. 6.15 provides additional insight towards the sequence of events which result in the creation/destruction of $\hat{\mathcal{C}}_B$ as ρ increases/decreases. Moving from left to right, these images show the result of the tracking procedure when tracking the evolution of $\hat{\mathcal{C}}_B$ as ρ decreases. As indicated by the dashed trajectory in the middle panel, $\hat{\mathcal{C}}_B$ is found to either become unstable or no longer exist between $\rho = 0.8937$ and 0.8936 and the state of the RC is on a transient to LC_1. The RC then remains on LC_1 when ρ is decreased further as illustrated by the image on the right for $\rho = 0.8935$.

The third row of images in Fig. 6.15 depicts the previously mentioned torus bifurcation and the subsequent death of this torus as ρ increases. By increasing ρ from 0.9073 to 0.9074, the torus can no longer be tracked. Following this bifurcation the state of the RC converges from the torus (as illustrated by the dashed line) to $\hat{\mathcal{C}}_B$ and, according to the tracking algorithm, remains on $\hat{\mathcal{C}}_B$ when ρ is increased as shown in the right panel for $\rho = 0.9075$.

The bottom row of images show that by increasing ρ from 0.9419 to 0.9420 LC_1 can no longer be tracked. As $\hat{\mathcal{C}}_B$ is the only remaining stable solution of the RC for $\rho = 0.9420$, the state of the RC has no choice but to embark on a transient trajectory from the previously stable LC_1 to $\hat{\mathcal{C}}_B$ (plotted as the dashed line) and remains on $\hat{\mathcal{C}}_B$ as ρ is increased to 0.9421.

The rise and fall of multifunctionality

The range of ρ values where the RC achieves a coexistence of $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ in $\mathbb{P}$ are also shown in Fig. 6.12. To expand on this, Fig. 6.16 shows how the rise and fall of multifunctionality begins with the arrival of $\hat{\mathcal{C}}_A$ at $\rho = 1.0831$ and death of $\hat{\mathcal{C}}_B$ at $\rho = 1.4297$.

The top panel of Fig. 6.16 picks up where Fig. 6.15 leaves off by continuing to track the evolution of $\hat{\mathcal{C}}_B$ with regard to changes in ρ. The second row of images in Fig. 6.16 highlights the circumstances in which $\hat{\mathcal{C}}_B$ can no longer be tracked. By increasing ρ from 1.4296 to 1.4297 it is shown here that $\hat{\mathcal{C}}_B$ no longer has a fixed amplitude of oscillation. Following this, by increasing ρ to 1.4298, $\hat{\mathcal{C}}_B$ is no longer a stable limit cycle and the state of the RC is instead on a transient to $\hat{\mathcal{C}}_A$ and remains on $\hat{\mathcal{C}}_A$ for increasing ρ.

The third row of images in Fig. 6.16 depicts how $\hat{\mathcal{C}}_A$ is born. Shown from left to right is the result of tracking the evolution of $\hat{\mathcal{C}}_A$ for decreasing ρ. Here it can be seen that between $\rho = 1.0831$ and 1.0830, $\hat{\mathcal{C}}_A$ can no longer be tracked and the state of the RC subsequently tends towards $\hat{\mathcal{C}}_B$ and remains on $\hat{\mathcal{C}}_B$ for smaller ρ values.

While the top panel of Fig. 6.7 shows an instance in which the RC fails to achieve multifunctionality for $\rho = 1.7$, the top panel of Fig. 6.16 reveals the means in which this comes about. Following the death of $\hat{\mathcal{C}}_B$ at $\rho = 1.4298$, $\hat{\mathcal{C}}_A$ is the only stable solution present in $\mathbb{P}$. However, $\hat{\mathcal{C}}_A$ is found to undergo a torus bifurcation at $\rho = 1.4711$. Following this, several windows of periodic and chaotic behaviour appear and the RC remains chaotic for $\rho > 1.6$. As ρ is increased beyond this point, the range of local maxima in the x variable, as plotted here, continues to grow. As a result, the dynamics of the RC loses its resemblance to the original driving input.

Furthermore, the bifurcation diagram presented in Fig. 6.16 provides additional information on the results shown previously in Fig. 6.7. Fig. 6.16 shows that the trajectories taken by the RC when initialised with either $\boldsymbol{r}_{\mathcal{C}_A}(t_{\text{listen}})$ or $\boldsymbol{r}_{\mathcal{C}_B}(t_{\text{listen}})$ for $\rho = 1.7$ are trajectories on the same chaotic attractor.

Floquet analysis

The results presented thus far in this section illustrate some of the hurdles which the RC overcomes in order to achieve multifunctionality. However, given the nature of the seeing double problem, it is also possible to provide some quantitative insight into how multifunctionality is achieved by conduct-

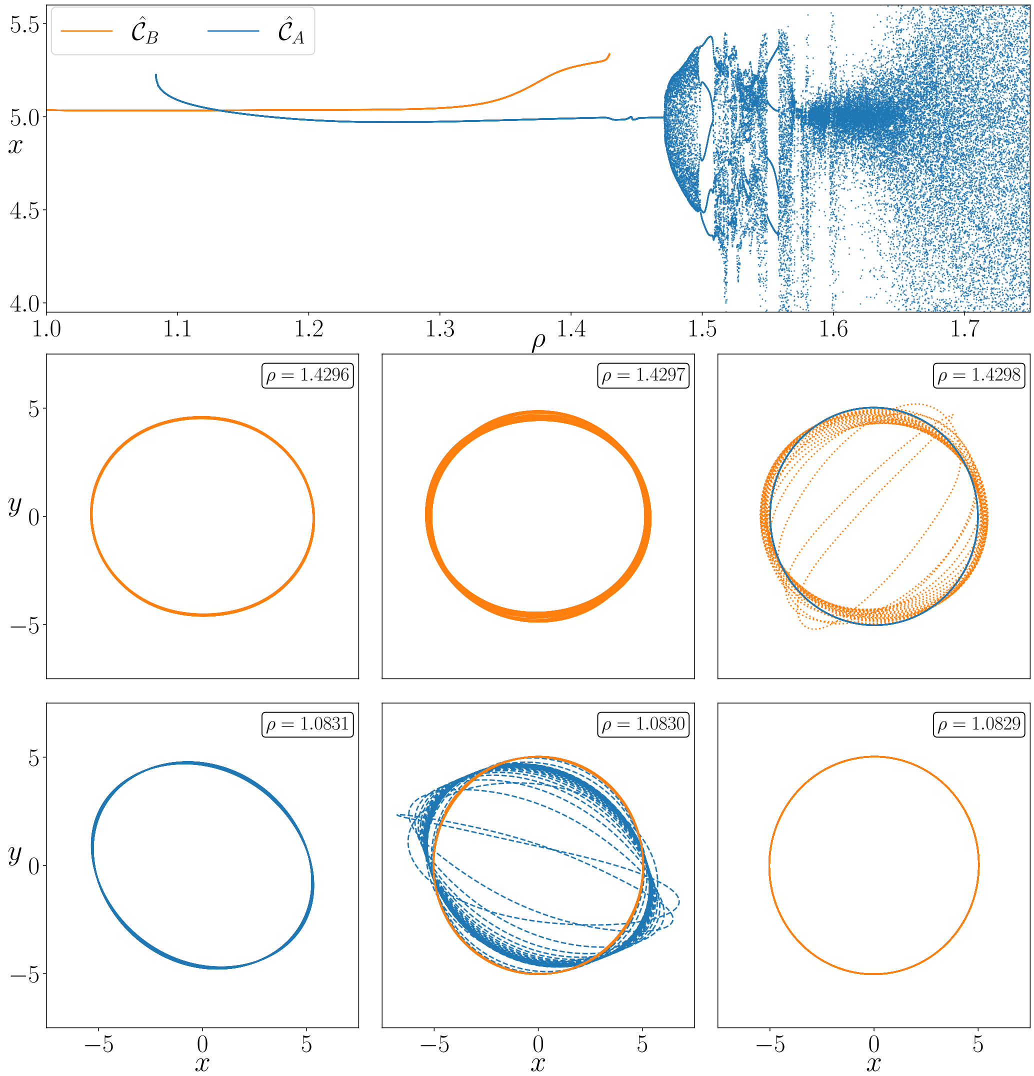


Figure 6.16: Top left panel illustrates how the local maxima of $\hat{\mathcal{C}}_B$ and $\hat{\mathcal{C}}_A$ change in the (ρ, x)-plane. The panels in the second and third row show show the dynamics of $\hat{\mathcal{C}}_B$ and $\hat{\mathcal{C}}_A$ respectively just before these attractors can no longer be tracked.

ing a Floquet analysis.

This involves computing the eigenvalues of the 'monodromy matrix', $\mathbf{Q}$, which is obtained by solving Eq. C.8. These eigenvalues, μ_i, are called the Floquet multipliers and the information which they reveal depends on the choice of driving periodic signal in Eq. C.8. The steps involved in determining the Floquet multipliers are outlined in Appendix C.2.

In the case of the problem at hand, by solving Eq. C.8 with input of one period from, for instance, $\hat{\mathcal{C}}_A$ then the largest multiplier, μ_1, will have magnitude $= 1$ and all other $|\mu_i| < 1$. While this provides information on when $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ are stable at certain ρ values and compliments results shown previously in the present Chapter, the particular bifurcation through which the attractors become stable/unstable can also be determined. The criteria under which each bifurcation is identified is outlined in Appendix C.2. Furthermore, dynamical features which the tracking algorithm fails to notice may be unearthed by conducting such an analysis.

Taking this into account, Fig. 6.17 shows how the 5 largest Floquet multipliers evolve with respect to changes in ρ when solving Eq. C.8 using a drive of one period on $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ in the top and bottom panels respectively. In each case, $|\mu_1| = 1$ at all ρ values where $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ were found to exist as stable attractors.

As ρ decreases, Fig. 6.17 shows that $\mathrm{Re}(\mu_2) \to 1$ and since all $\mathrm{Im}(\mu_i) = 0$ it can be deduced that $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ are born through a saddle-node bifurcation.

For $\hat{\mathcal{C}}_B$, Fig. 6.17 demonstrates that $\mathrm{Re}(\mu_2)$ and $\mathrm{Re}(\mu_3)$ become equal at $\rho = 1.3690$ and beyond this point the corresponding imaginary components of μ_2 and μ_3 grow according to $\mathrm{Im}(\mu_2) = -\,\mathrm{Im}(\mu_3)$ until $|\mu_2|$ and $|\mu_3|$ tend to 1 resulting in $\hat{\mathcal{C}}_B$ becoming unstable at $\rho = 1.4927$. Based on this information and the results shown in Fig. 6.16, Fig. 6.17 indicates that $\hat{\mathcal{C}}_B$ becomes unstable through a subcritical torus bifurcation.

The behaviour of the Floquet multipliers computed for $\hat{\mathcal{C}}_A$ in Fig. 6.17 establishes that a supercritical torus bifurcation is responsible for the $\hat{\mathcal{C}}_A$'s change in dynamics from stable limit cycle to stable torus at $\rho = 1.4711$ as illustrated in Fig. 6.16

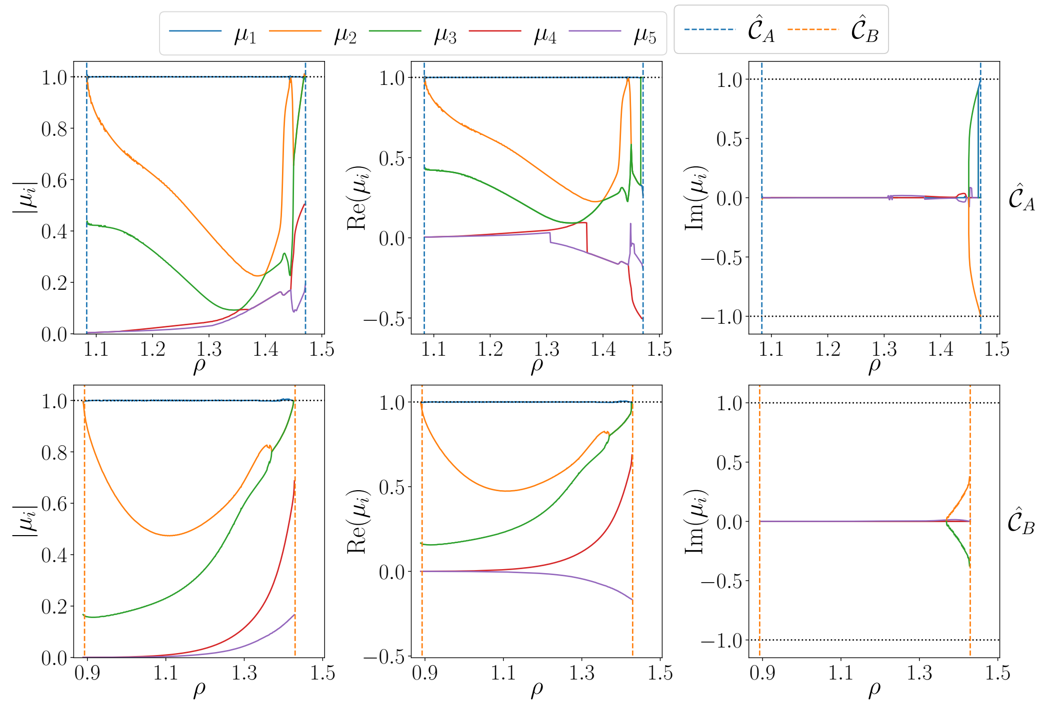


Figure 6.17: Changes in the magnitude, real part, and imaginary part of the first $i = 5$ largest Floquet multipliers, μ_i, with respect to changes in ρ when driving the RC with input from $\hat{\mathcal{C}}_A$ (top row) and $\hat{\mathcal{C}}_B$ (bottom row). Dashed vertical lines highlight the ρ values where $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ were found to exist between.

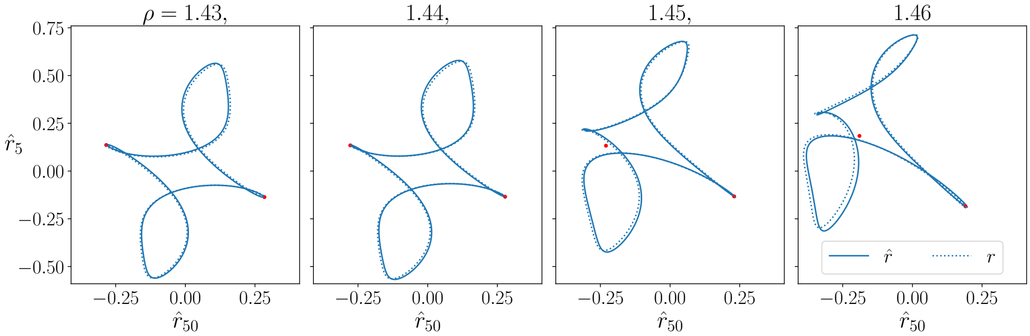


Figure 6.18: RC dynamics in the $(\hat{r}_5, \hat{r}_{50})$-plane nearby a symmetry breaking bifurcation. Red points depict the location of $(\hat{r}_5(t), \hat{r}_{50}(t))$ and $(-\hat{r}_5(t), -\hat{r}_{50}(t))$ at a time t. Dashed curves illustrate the corresponding dynamics of the RC in training.

The Floquet multipliers also reveal further information about the RC's dynamics which Fig. 6.16 fails to capture. A symmetry breaking bifurcation on $\hat{\mathcal{C}}_A$ is found to occur at $\rho = 1.4450$ as $\mathrm{Re}(\mu_2)$ sharply increases to 1 and quickly decreases from 1. Fig. 6.18 illustrates the dynamics of some of the RC's individual neurons, specifically $\hat{r}_5$ and $\hat{r}_{50}$, nearby this bifurcation in order to highlight the emergence of symmetry broken states. Prior to the bifurcation, Fig. 6.18 shows that for $\rho = 1.43$ and 1.44, $(\hat{r}_5(t), \hat{r}_{50}(t)) = (-\hat{r}_5(t), -\hat{r}_{50}(t))$ $\forall\, t$. However, after the bifurcation this relationship no longer holds as seen here for $\rho = 1.45$ and 1.46.

In light of these results, it now becomes apparent that the small but noticeable reappearance of $\mathbf{W}_{out}^{(2)}$ in Fig. 6.10a occurs due to this symmetry breaking bifurcation and its rebirth is not solely a feature of the RC exhibiting chaos. At the same time, the magnitude of $\mathbf{W}_{out}^{(2)}$'s elements begin to grow rapidly once the RC exhibits chaos.

The dynamics of the same neurons, $\hat{r}_5$ and $\hat{r}_{50}$, during the training stage are also plotted in Fig. 6.18 for a given ρ. The behaviour of these neurons demonstrates that the symmetry breaking bifurcation at $\rho = 1.445$ does not violate the criteria outlined in Sec. 5.2 which describes the circumstances that enables the 'symmetry kills the square' phenomenon to occur. As a further note, Fig. 6.18 also shows that as ρ increases, $\hat{r}_5$ and r_5 behave less like one another, and the same is said for $\hat{r}_{50}$ and r_{50}.

6.5.2 Bifurcation Analysis in the (x_{cen}, ρ)-plane

The results presented in Sec. 6.5.1 provide an extensive account of the particular bifurcations the RC goes through in order to achieve multifunctionality when $x_{cen} = 0$. These results suggest that for increasing ρ, on one hand, the dynamics of FP_1, FP_2, FP_3, and FP_4 which lead to the creation of LC_1 and LC_2, symbolise a set of challenges for the RC to overcome in order to achieve multifunctionality. On the other hand, as ρ increases it can also be argued that the role these fixed points and limit cycles play is to sculpt $\mathbb{P}$ for the arrival of $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$.

At the same time, what these results also bring into focus is the strong level of interaction between these fixed points and limit cycles as Figs. 6.12 and 6.13 show that these six attractors coexist for a small range of ρ values when $x_{cen} = 0$. In this section many of the same tools that were applied in Sec. 6.5.1 are used to explore the dynamics of the RC in the (x_{cen}, ρ)-plane.

The results presented in this section reveals how the relationships between the fixed points and limit cycles mentioned above evolve with respect to changes in x_{cen}, and provides insight on the different roads to $\mathcal{C}_A$ and $\mathcal{C}_B$'s reconstruction as conceivably carved out by these fixed points and limit cycles.

Tracking stable solutions in the (x_{cen}, ρ)-plane

The result of tracking the changes in the dynamics of the RC's stable solutions for $\rho \in [0.48, 0.52]$ and $x_{cen} \in [0.575, -0.575]$ are presented in Fig. 6.19. These results are generated by tracking the evolution of LC_1 and LC_2 for changes in x_{cen} beginning nearby the bifurcations at $\rho = 0.4847$ and 0.4905, which spawn LC_1 and LC_2 respectively when $x_{cen} = 0$, as illustrated in Figs. 6.12 and 6.13. Similarly, the evolution of FP_1, FP_2, FP_3, and FP_4 are also tracked with respect to changes in x_{cen} nearby the bifurcations at $\rho = 0.4935$ and 0.4931 resulting in the death of these fixed points. Once these attractors can no longer be tracked for changes in x_{cen}, this location in the (x_{cen}, ρ)-plane is recorded and added to the corresponding curve in Fig. 6.19. The same process is repeated for incremental changes in ρ for $\rho \in [0.48, 0.52]$ and for regions where $\hat{\mathcal{C}}_A$ or $\hat{\mathcal{C}}_B$ are encountered.

The labels and colour scheme of each curve in Fig. 6.19 are consistent with illustrations of the attractors studied in the previous section. While, for example, the stable period-1 limit cycle, LC_1, is found to undergo period-doubling bifurcations at several locations in the $(x_{cen},\, \rho)$-plane, the resulting stable period-2 limit cycle is also referred to as LC_1. This convention, which was also taken in relation to LC_2 and T_1 in Fig. 6.15, is applied even when, for instance, an infinite sequence of period-doubling bifurcations have taken place when tracking the dynamics of the original period-1 limit cycle.

The colour scheme of the filled regions in the portion of the $(x_{cen},\, \rho)$-plane shown in Fig. 6.19, as defined by the area the above curves enclose, describe the nature of the RC's stable solutions in $\mathbb{P}$ for a given x_{cen} and ρ.

Furthermore, what is not shown nor highlighted along the curves mentioned above in Fig. 6.19 are the corresponding codim-1 curves and codim-2 bifurcations which take place on certain attractors in the $(x_{cen},\, \rho)$-plane. This decision was taken in order to avoid overcrowding Fig. 6.19 with excessive information, which may diminish the impact of its main message. Instead, a more detailed breakdown on some of the bifurcations which take place in Fig. 6.19 is provided later in the present section.

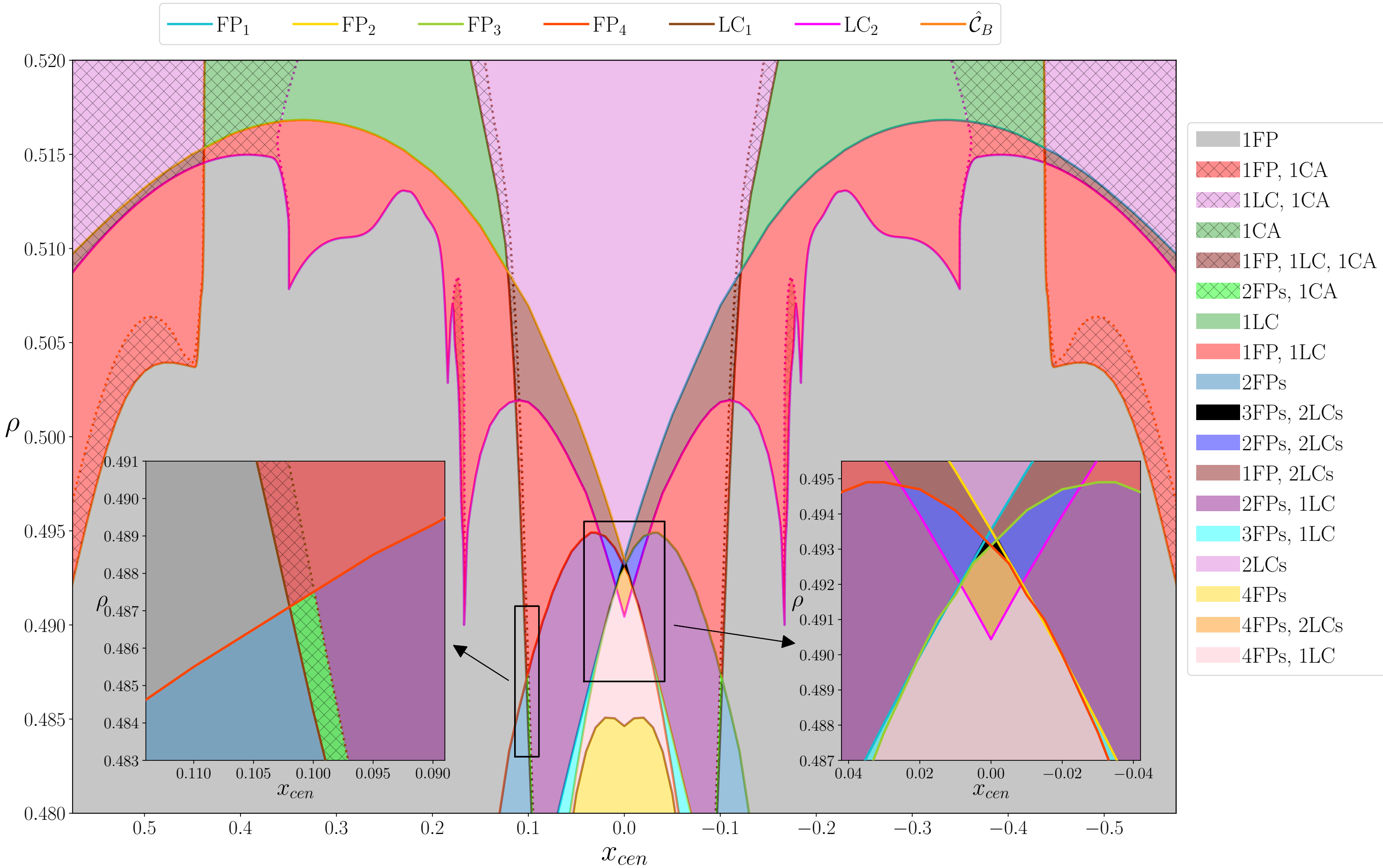


Figure 6.19: Result of tracking the evolution of stable attractors in $\mathbb{P}$ for $\rho \in [0.48, 0.52]$ and $x_{cen} \in [0.575, -0.575]$. Curves represent the end point of tracking stable solutions. Filled regions describe the nature of the RC's stable solutions.

General comments on RC's dynamics in the (x_{cen}, ρ)-plane

The primary message of Fig. 6.19 continues the central narrative of this Chapter, the greater the overlap between $\mathcal{C}_A$ and $\mathcal{C}_B$, the more memory the RC requires in order to achieve multifunctionality and the greater the influence of these fixed points and limit cycles until reconstruction occurs.

In Fig. 6.19 the above statement is further substantiated through examination of the behaviour of the stable solutions in $\mathbb{P}$ for the selected x_{cen} and ρ values. What this reveals is the wider role these limit cycles, and the fixed points which create them, play in terms of how the RC solves the seeing double problem.

A detailed account on the behaviour and bifurcations of each of the RC's stable solutions in $\mathbb{P}$, for the specified range of x_{cen} and ρ values in Fig. 6.19, is outlined throughout the remainder of the present section.

However, before discussing the dynamics of each of the stable solutions in Fig. 6.19, it is important to note that one of the main features of Fig. 6.19 is the symmetry in the RC's dynamics about $x_{cen} = 0$. In the case of each fixed point, for a given pair of (x_{cen}, ρ), the corresponding mirrored fixed point exist for ($-x_{cen}$, ρ). Furthermore, in the case of limit cycle and chaotic attractor, the same attractor exists for each $\pm x_{cen}$. This behaviour comes as another corollary of the results regarding symmetry outlined in Sec. 6.4.4.

Behaviour of FP_3 and FP_4

Concentrating first on FP_3 and FP_4, Fig. 6.19 reveals that as the magnitude of x_{cen} increases these fixed points play a diminishing role in the RC's dynamics.

Fig. 6.19 shows that at $\rho = 0.48$, these fixed points cease to coexist for $|x_{cen}| > 0.055$ and both no longer exist for $|x_{cen}| > 0.13$.

The inset plot on the right hand side of Fig. 6.19 highlights that for increasing ρ along the FP_3 and FP_4 curves from $x_{cen} = 0.13$ and -0.13 respectively, FP_3 and FP_4 become single point-like branches at $x_{cen} \approx -0.035$ and 0.035 respectively and can no longer be tracked for $\rho > 0.495$.

Behaviour of FP_1 and FP_2

Fig. 6.19 reveals that as the magnitude of x_{cen} increases, FP_1 and FP_2 begin to play a more prominent role in the RC's dynamics in comparison to FP_3 and FP_4 for the particular ρ values explored here in the $(x_{cen},\ \rho)$-plane.

While Fig. 6.19 shows that at $\rho = 0.48$, the coexistence of FP_1 and FP_2 comes to an end for $|x_{cen}| > 0.07$, these fixed points exist far beyond the limits of the x_{cen}-axis displayed here. What is more is that, together, these fixed points occupy the largest combined area in the portion of the $(x_{cen},\ \rho)$-plane shown here. For increasing ρ along the curves which enclose the regions of the $(x_{cen},\ \rho)$-plane where FP_1 and FP_2 are both stable, the same fate as FP_3 and FP_4 awaits FP_1 and FP_2 as these become single point-like branches at $x_{cen} \approx 0.34$ and -0.34 respectively.

As a further comment, the inset plot on the right hand side of Fig. 6.19 provides additional insight to the bifurcation diagrams in Figs. 6.12 and 6.13 and draws attention to how the coexistence between FP_1 and FP_2, and FP_3 and FP_4 at $x_{cen} = 0$ unfolds with respect to changes in x_{cen}.

Behaviour of LC_1

Focusing next on LC_1, Fig. 6.19 illustrates the limited role this limit cycle plays in the RC's dynamics as the magnitude of x_{cen} increases from 0. For decreasing ρ, the range of x_{cen} values where LC_1 is stable also decreases. By increasing/decreasing x_{cen} from 0 for $\rho = 0.48$, a region in the $(x_{cen},\ \rho)$-plane where all four fixed points coexist, LC_1 is first encountered as a stable solution at $x_{cen} \approx \pm 0.05$. LC_1 briefly coexists with these four fixed points until FP_3 and FP_4 can no longer be tracked as previously mentioned at $x_{cen} = \mp 0.055$ and likewise for FP_1 and FP_2 at $x_{cen} = \pm 0.07$. This leaves LC_1 in coexistence with FP_2 and FP_3 for $x_{cen} \in [0.071, 0.097]$ and in coexistence with FP_1 and FP_4 for $x_{cen} \in [-0.071, -0.097]$. LC_1 can no longer be tracked for $|x_{cen}| > 0.097$ at $\rho = 0.48$.

As a further comment, as ρ decreases, the brown curves on either side of $x_{cen} = 0$, which outline where LC_1 is stable, appear to be converging to a point along the FP_3 and FP_4 curves. This suggests that a codim-2 bifurcation

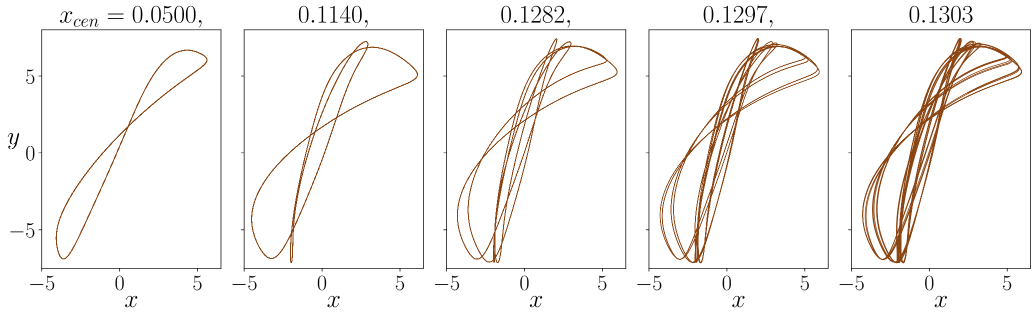


Figure 6.20: Illustrating the evolution of LC_1 in $\mathbb{P}$ with respect to changes in x_{cen} for $\rho = 0.515$. From left to right, each image depicts LC_1 at specified x_{cen} values after successive period-doubling bifurcations which gives rise to chaos.

takes place in the $(x_{cen},\ \rho)$-plane outside of the regions explored in Fig. 6.19.

Prior to no longer being able to track LC_1 for increasing the magnitude of x_{cen}, Fig. 6.19 also illustrates that LC_1 begins to exhibit chaos along the dotted brown curve $\forall\, \rho$ values shown here. The extent of this chaos in the $(x_{cen},\ \rho)$-plane is described by the area enclosed by this dotted curve and the LC_1 curve, and is further emphasised by the hatched pattern. It is shown here that, on one hand, these windows of chaos widen as ρ increases, on the other hand, the inset plot on the left hand side of Fig. 6.19 provides a more detailed picture on how, this small region of chaos narrows as ρ decreases.

Fig. 6.20 sheds some light on the change from periodic to chaotic dynamics, as characterised by this dotted curve, when tracking the evolution of the original stable period-1 limit cycle. For $\rho = 0.515$, it is shown here that as x_{cen} increases, LC_1 undergoes a series of period-doubling bifurcations. An infinite sequence of these bifurcations occurs between $x_{cen} = 0.1301$ and 0.1302 which gives rise to the chaotic dynamics shown in Fig. 6.20 for $x_{cen} = 0.1303$.

Behaviour of LC_2

Fig. 6.19 lays bare the intricate relationship between LC_2, ρ, and x_{cen}. As the magnitude of x_{cen} increases several events occur along the LC_2 curve. The inset plot on the right hand side highlights the first of these incidents with the loss of coexistence between the four fixed points, LC_1, and LC_2. This continues with larger $|x_{cen}|$ values and at $\rho = 0.502$, LC_2 is in coexistence with only FP_1 for $x_{cen} < -0.115$ and FP_2 for $x_{cen} > 0.115$ until $\mathcal{C}_B$ is reconstructed at larger x_{cen} values.

It is also shown in Fig. 6.19 that nearby these points in the $(x_{cen},\ \rho)$-plane where LC_2 loses its coexistence with LC_1, the LC_2 curve itself reaches its first of several local maxima. In order to continue tracking LC_2 beyond this point, ρ is required to decrease as $|x_{cen}|$ increases until the curve reaches its first of several local minima at $(x_{cen},\ \rho) = (\pm 0.16, 0.49)$ where a period-doubling bifurcation occurs. Subsequent period-doubling bifurcations are encountered along the LC_2 curve for slightly increasing $|x_{cen}|$ and ρ. The hatched region of the (x, ρ)-plane enclosed by the LC_2 curve and the nearby dotted curve characterise the small window of chaotic solutions produced by an infinite sequence of these period-doubling bifurcations.

Fig. 6.21 provides an example of the RC's dynamics nearby one of these period-doubling cascades for $\rho = 0.501$. It is shown here that when tracking the evolution of the original stable period-1 limit cycle, successive period-doubling bifurcations are encountered for increasing x_{cen}. Fig. 6.21 shows examples of the corresponding period-1 dynamics at $x_{cen} = 0.1500$, period-2 dynamics at $x_{cen} = 0.1660$, and period-4 dynamics at $x_{cen} = 0.1665$. In this scenario the period-8 limit cycle, whose dynamics are shown here for $x_{cen} = 0.1667$, undergoes an infinite sequence of period-doubling bifurcations for a small increase in x_{cen}. This produces the chaotic dynamics shown here for $x_{cen} = 0.1669$.

This region of chaos in Fig. 6.19 ends once the LC_2 curve and dotted curve meet at $(x_{cen}, \rho) = (0.178, 0.507)$. For increasing $|x_{cen}|$ beyond this point, it is found that LC_2 is a period-2 limit cycle before it can no longer be tracked. However, subsequent period-doubling bifurcations occur along the LC_2 curve for increasing $|x_{cen}|$.

Following an infinite sequence of these period-doubling bifurcations, what Fig. 6.19 also shows is that along the LC_2 curve, its dynamics become chaotic again at $(x_{cen},\ \rho) = (\pm 0.35, 0.508)$. The dotted purple curve which stems from this point distinguishes between when LC_2 is periodic or chaotic as further

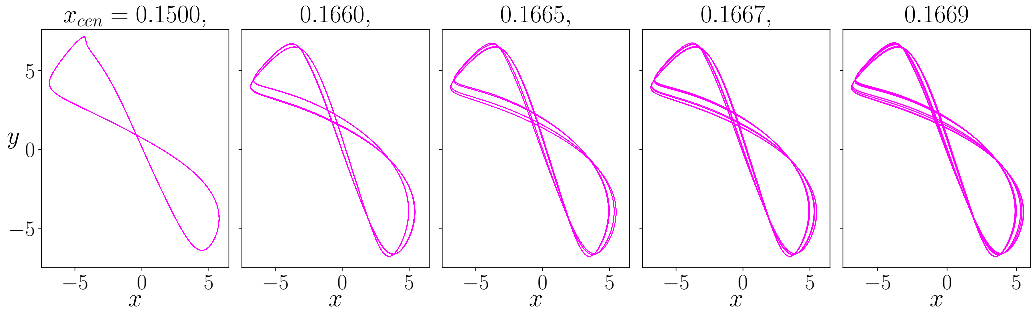


Figure 6.21: Illustrating the evolution of LC_2 in $\mathbb{P}$ with respect to changes in x_{cen} for $\rho = 0.501$. From left to right, each image depicts LC_2 at specified x_{cen} values after successive period-doubling bifurcations which gives rise to chaos.

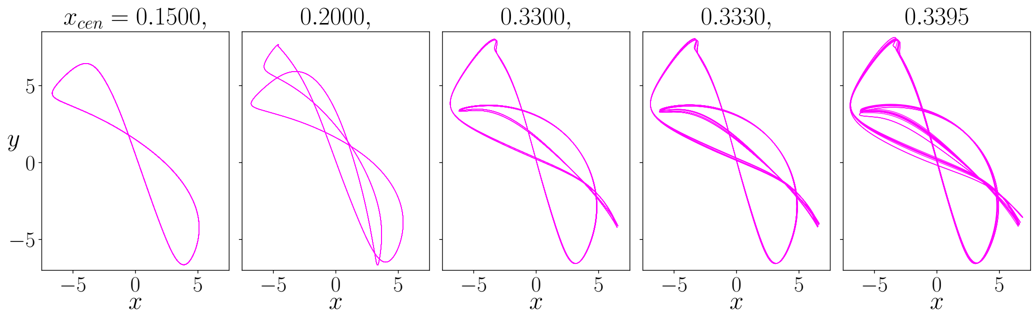


Figure 6.22: Illustrating the evolution of LC_2 in $\mathbb{P}$ with respect to changes in x_{cen} for $\rho = 0.511$. From left to right, each image depicts LC_2 at specified x_{cen} values after successive period-doubling bifurcations which gives rise to chaos.

emphasised by the hatched pattern. Furthermore, although it is not shown here, these period-doubling bifurcation do not only occur along the LC_2 curve, there is a locus which connects each given set of period-doubling bifurcations points in the (x_{cen}, ρ)-plane in a similar way to the dotted purple curve mentioned above. An example of this particular route to chaos is provided in Fig. 6.22.

What Fig. 6.22 depicts is the result of tracking the evolution of LC_2's dynamics for changes in x_{cen} when $\rho = 0.511$. The remnants of the previous small window of chaos, which occurs after the first local minima along the LC_2 curve, are visible here with a period-2 limit cycle found for $x_{cen} = 0.2000$. The major difference between this period-2 limit cycle and the example illustrated in Fig 6.21 for $\rho = 0.501$ is the prominence of the sharp turn around the max y value on the period-2 limit cycle in Fig. 6.22. This remains a distinctive feature of LC_2 when tracking its dynamics as x_{cen} is further increased and persists after each of the subsequent period-doubling bifurcations which give rise to chaos. It is also worth noting that this kink along LC_2's dynamics occurs nearby the location of FP_2 and FP_1 for the corresponding positive and negative x_{cen} values respectively.

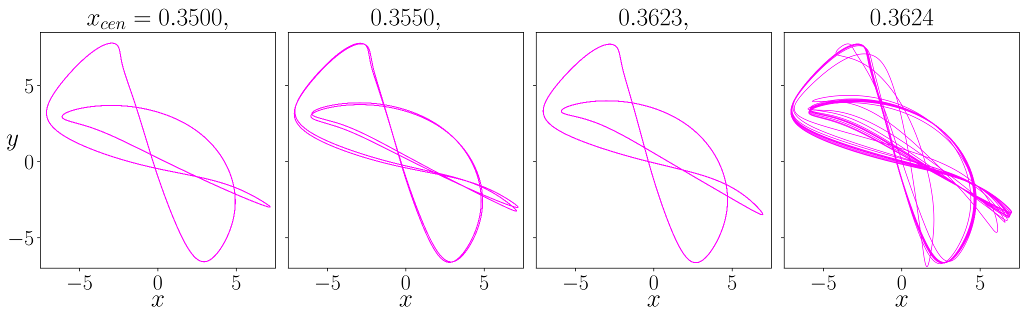
LC_2 remains chaotic as $|x_{cen}|$ increases from 0.35 and, depending on the sign of x_{cen}, continues its coexistence with either FP_1 or FP_2 for the range of x_{cen} values shown in Fig. 6.19. Much like LC_1 with FP_3 and FP_4, Fig. 6.19 also suggests that the LC_2 curve may connect with the FP_1 and FP_2 curves beyond the range of x_{cen} values shown here.

As ρ increases along this dotted curve of chaos, which begins at $(x_{cen},\, \rho) = (\pm 0.35, 0.508)$, there is a point where these sequences of period-doubling bifurcations are no longer directly responsible for the onset of chaotic dynamics. For instance, Fig. 6.23a shows that when tracking the evolution of LC_2 for $\rho = 0.515$, by increasing x_{cen} from 0.3623 to 0.3624, there is a sudden change in dynamics from a period-2 limit cycle to a chaotic attractor. The same is shown Fig. 6.23b for $\rho = 0.517$ and x_{cen} increases from 0.3558 to 0.3559. The only difference between these examples of LC_2's dynamics prior to the dawning of chaos is the lack of a period-4 limit cycle for $\rho = 0.517$.

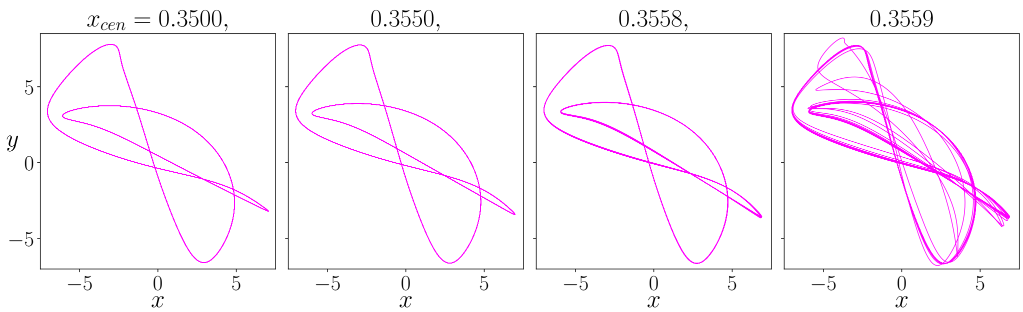
Behaviour of $\hat{\mathcal{C}}_B$

Lastly, Fig. 6.19 also illustrates regions in the $(x_{cen},\, \rho)$-plane where $\hat{\mathcal{C}}_B$ comes into existence. For much of its presence in Fig. 6.19, $\hat{\mathcal{C}}_B$ is in coexistence with either, LC_2, or one of FP_2 or FP_1 depending on the sign of x_{cen}, or both LC_2 and one of these fixed points.

At the limits of the x_{cen}-axis in Fig. 6.19, i.e., when $|x_{cen}| = 0.575$, the nature of $\hat{\mathcal{C}}_B$ differs greatly depending on ρ. In particular, the dotted orange curve describes the boundary between when $\hat{\mathcal{C}}_B$ exhibits periodic or chaotic behaviour, and it is within the hatched region that $\hat{\mathcal{C}}_B$ is chaotic. For decreasing $|x_{cen}|$, these two curves approach one another and the window of chaos



(a) $\rho = 0.515$



(b) $\rho = 0.517$

Figure 6.23: Illustrating the evolution of LC_2 in $\mathbb{P}$ with respect to changes in x_{cen} for $\rho = 0.515$ in (a) and $\rho = 0.517$ in (b). From left to right, each image depicts LC_2 at specified x_{cen} values.

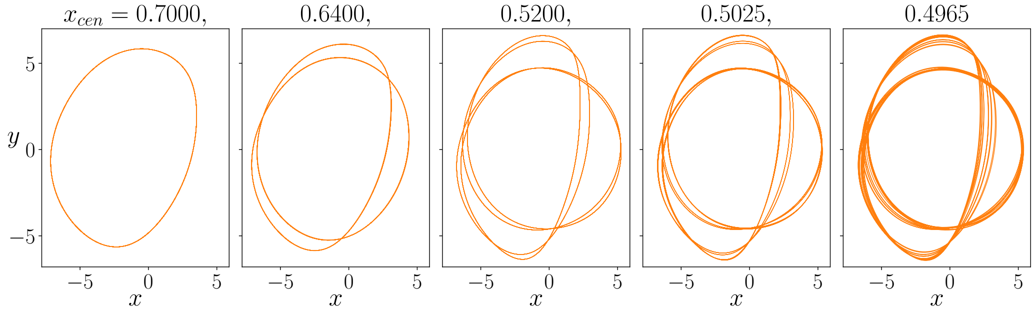


Figure 6.24: Illustrating the evolution of $\hat{\mathcal{C}}_B$ in $\mathbb{P}$ with respect to changes in x_{cen} for $\rho = 0.504$. From left to right, each image depicts $\hat{\mathcal{C}}_B$ at specified x_{cen} values after successive period-doubling bifurcations which gives rise to chaos.

becomes arbitrarily small at $|x_{cen}| = 0.45$. Furthermore, it is from this point in the $(x_{cen},\ \rho)$-plane that by slightly decreasing $|x_{cen}|$ a sudden increase in ρ is required in order to reconstruct $\hat{\mathcal{C}}_B$.

Fig. 6.24 illustrates the changes in $\hat{\mathcal{C}}_B$'s dynamics when tracking its evolution for decreasing x_{cen} when $\rho = 0.504$. Here it can be seen that $\hat{\mathcal{C}}_B$ is a period-1 limit cycle for $x_{cen} = 0.7000$ and is also not perfectly circular. $\hat{\mathcal{C}}_B$ undergoes several period-doubling bifurcations with examples of period-2, period-4, and period-8 dynamics shown here for $x_{cen} = 0.6400, 0.5200$, and 0.5205. An infinite sequence of period-doubling bifurcations occurs as x_{cen} is decreased further resulting in the chaotic dynamics like those shown here for $x_{cen} = 0.4965$.

Above all, what Figs. 6.24 and 6.19 show is that while $\hat{\mathcal{C}}_B$ is stable its dynamics do not always resemble $\mathcal{C}_B$ even to the point when $\hat{\mathcal{C}}_B$ is not in any way circular, has a much larger period, or even exhibits chaos. At the same time, it is also worth noting that after a number of period-doubling bifurcations, for instance the period-4 limit cycle shown in Figs. 6.24, portions of the trajectory on $\hat{\mathcal{C}}_B$ begin to resemble a more circular orbit with similar characteristics to $\mathcal{C}_B$. This suggests that these period-doubling bifurcations may play an important role in how the RC improves its reconstruction of $\mathcal{C}_B$.

6.6 Conclusions

While it is known from previous Chapters that RCs can exhibit multifunctionality in exotic cases involving multiple overlapping chaotic attractors, these examples make it difficult to 'open the black-box' and understand the fundamentals of how the RC achieves multifunctionality in the first place. These examples also make it more challenging to address the complications which arise due to overlapping data as discussed in Sec. 6.1.

The results presented in Sec. 6.2 make some progress on addressing the above issues by examining the difficulties encountered by different RC designs when tasked with reconstructing a coexistence of chaotic attractors which are moved closer together. However, there are certain limitations to what can be inferred about multifunctionality in a more general sense with this approach. It is by introducing the issue of overlapping training data via the the seeing double problem in Sec. 6.3 that the results presented in this Chapter help to provide a deeper understanding of how RCs achieve multifunctionality and overcome the problems posed by overlap.

The results presented in this Chapter show that in order for the RC to achieve multifunctionality there is a crucial dependence on ρ, the spectral radius of the RC's internal connections. The intricacies of this relationship are revealed in Sec. 6.4.1 when the cycles $\mathcal{C}_A$ and $\mathcal{C}_B$ are moved closer together. As $\mathcal{C}_A$ and $\mathcal{C}_B$ begin to overlap, Fig. 6.4 shows that the RC requires a greater amount of memory in order to successfully distinguish between the orbits. A 'Goldilocks' effect is also found to occur whereby if ρ is too small or too large then the RC will fail to learn.

Sec. 6.4.3 focuses on how the RC constructs the basins of attraction for $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ when they are entirely overlapping from the perspective of $\mathbb{P}$.

The nature of $\mathbb{P}$ prior to the RC achieving multifunctionality is also explored. Fig. 6.9 unveils the coexistence of several fixed points, limit cycles and tori in $\mathbb{P}$, attractors which were not involved in the training. This discovery provides the basis for conducting the bifurcation analyses carried out in Sec. 6.5 in order to determine the role these attractors play in how the RC solves the seeing double problem.

This bifurcation analysis is divided into two studies, the results of the first are outlined in Sec. 6.5.1 which explores how changes in ρ effects the RC's dynamics in the case of $x_{cen} = 0$. The results of the second study are presented in Sec. 6.5.2 where the RC's dynamics in a portion of the (x_{cen}, ρ)-plane are assessed. Here the role of these fixed points and limit cycles in the learning dynamics of the RC becomes apparent. The particular bifurcations which result in the rise and fall of multifunctionality are identified through the results of the Floquet analysis shown in Fig. 6.17. The main take home message from both of these bifurcation analyses is the same, the more $\mathcal{C}_A$ and $\mathcal{C}_B$ overlap, the more memory the the RC requires in order to exhibit multifunctionality and the greater the influence of these fixed points and limit cycles until reconstruction occurs.

Despite the insight gained from the results presented in this Chapter, there are still many unanswered questions concerning the interesting dynamics a RC can be trained to produce. For instance, how much multifunctionality can a RC have? Are there certain network topologies or RC formulations which enhance multifunctionality? From an applications perspective, how can multifunctionality be exploited as a data-driven modelling tool?

Chapter 7

Limits of Multifunctionality

In this Chapter the extreme case of the seeing double problem studied in the previous Chapter is reinterpreted as a benchmark task for exploring the limits of multifunctionality across different RCs. The relationship between multifunctionality and several RC parameters are assessed. These results reveal when certain RCs reach their optimal performance in this simplest but hardest case of overlapping training data, thereby providing guidance on how to train RCs to achieve multifunctionality in challenging circumstances.

7.1 Seeing Double Benchmarking

It is clear from the results presented in the previous Chapter that while the seeing double problem is a tough task to solve it provides a straightforward framework to understand the principles of how a RC achieves multifunctionality. The extreme case of the seeing double problem, as it is referred to throughout the previous Chapter, encapsulates the issues of overlapping training data in a paradigmatic setting which epitomises the difficulties posed by overlap in the hardest but simplest of scenarios.

Moreover, the seeing double problem provides elucidation on what are some of the necessary attributes a RC must possess in order to become multifunctional. One of the most highly influential of these is the relationship between multifunctionality and ρ. In the case where $\mathcal{C}_A$ and $\mathcal{C}_B$ are completely overlapping (i.e., when $x_{cen} = 0$), Fig. 6.4 provides some insight on how the limits of the relationship between multifunctionality and ρ are tested as multifunctionality is achieved for only a relatively small range of ρ values.

From a different perspective this task, of training a RC to reconstruct a coexistence of $\mathcal{C}_A$ and $\mathcal{C}_B$ when $x_{cen} = 0$, provides a suitable testing ground to assess what are the essential ingredients for a RC to become multifunctional.

Taking all of the above into consideration, in this Chapter the extreme case of the seeing double problem is used as a benchmark task to explore the limits of multifunctionality across different RCs.

Given the success of the prediction analysis procedure outlined in Sec. 6.4.1 which reliably detects whether multifunctionality was achieved (application of this method shown in Fig. 6.4), another aim of this Chapter is to take this analysis a step further by investigating the suitability of different tools to determine whether a given RC can achieve multifunctionality without explicitly examining the nature of the reconstructed attractors.

7.1.1 Exploring the limits of multifunctionality

In order to gain a greater understanding of the limits of multifunctionality, in this Chapter, the critical effects that certain parameters have in the training of different types of RCs are assessed when tasked with solving the seeing double problem. Furthermore, it is by exploring these limitations that advancements are made towards establishing the extent of the dynamical functionality a given RC can be trained to harness. This Chapter explores some of the differences in the performances of the continuous-time (CT), leaky-integrator (LI), and next generation (NG) RCs, described in Eqs. (2.10), (2.21), and (2.25) respectively[1].

The training data is generated according to the steps outlined in Sec. 6.3.1. Each of these RC setups are trained according to the blending technique from Sec. 3.2.1. The respective training data matrices and corresponding $\mathbf{W}_{out}$ are computed with the blending parameter, α, set at 0.5.

What is common amongst the CT and LI-RC setups is the random realisation of the input and internal connection matrices, $\mathbf{W}_{in}$ and $\mathbf{M}$. A key aspect of the studies conducted in Sec. 7.2 is the exploration of the issues which arise

[1]Most of the results presented in this Chapter are published in [33], Oliver Heilmann and Daniel Köglmayr conducted the experiments regarding the LI-RC and NG-RC respectively.

due to this element of randomness in both of these RC setups. In the case of the CT-RC, the relationship between randomness, ρ, N, and γ are assessed. Similarly, for the LI-RC setup, the relationship between randomness, ρ, and η are explored. As all RCs are trained using Eq. 2.16, in Sec. 7.2 the performance of each RC is assessed at different values of β and compared with one another.

7.1.2 Analysis Tools

In Sec. 7.2, the following analysis tools are used to assess the performance of the different RCs when tasked with solving the seeing double problem in the case of $x_{cen} = 0$.

Roundness criteria

To determine whether a given RC achieves multifunctionality, a similar roundness analysis procedure described at the beginning of Sec. 6.4.1 is used here. To assess how well a given RC reconstructs both cycles, the first step is to determine if the prediction of a given cycle is indeed periodic, and then if it is rotating in the correct direction. Following this the roundness is computed as the difference between the radius of the largest and smallest circle needed to enclose and inscribe the predicted cycle. If the maximum roundness of both roundness values is less than a threshold value $= 0.25$ (determined from empirical testing), then the RC has achieved multifunctionality.

At the same time there is a level of arbitrariness associated to the above approach as it requires a number of trail runs to determine an appropriate threshold roundness value. However, once a suitably robust threshold is identified this approach is highly reliable and relatively computationally cheap which therefore enables mass investigations on the relationship between multifunctionality and randomness in a given RC's design.

Short Term Memory capacity

As the particular data points which describe $\mathcal{C}_A$ and $\mathcal{C}_B$ are effectively the same, it is plausible to consider that the RC needs to have a sufficient 'memory' of its previous states in order to remain on the correct circular orbit and preserve the sense of rotation. The 'short-term memory' (STM) metric was proposed by Jaeger in [149] and is used to compute the 'memory capacity' of a given RC. This metric is computed without involving any information on the RC's performance of a given task. For the purposes of this study, the STM is calculated to assess if, in this sense, a given RC's memory is a valid indicator of whether the RC achieves multifunctionality.

The STM characterises the capability of a RC to remember past inputs. It is measured by training the RC to fit an input signal, $\nu[n]$, at time n to its time shifted signal, $\nu[n-j]$, with each point in ν chosen from a uniformly random, independent, and identically distributed set of real numbers in $[-1, 1]$. Instead of 'closing the loop' after training, the RC is driven with the input signal ν. The STM is then calculated as the square of the correlation between the RC output $\mathbf{W}_{out}^{j} r[n]$ and the true values given by $\nu[n-j]$ summed over all j,

$$\text{STM} = \sum\nolimits_j cor^2\Big(\nu[n-j], \mathbf{W}_{out}^{j} r[n]\Big). \tag{7.1}$$

W_{out}^{j} depends on j since for every j the training process needs to be repeated. Note that, $\mathbf{M}$, is shared by the RC used during prediction and the RC used to measure the STM while the input and output matrices are different.

It is important to note that this concept of short term memory is different to how memory has been discussed in relation to the spectral radius, ρ, so far in this Thesis.

Floquet analysis

In the previous Chapter, the result of computing the Floquet multipliers presented in Fig. 6.17 show precisely the range of ρ where the reconstructed attractors, $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$, are stable. These results also provide clarity on the bifurcations responsible for the birth and death of both $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$.

In this Chapter, the same procedure from Appendix C.2 which generated the results discussed above is repeated but instead of using input from $\hat{\mathcal{C}}_A$ and $\hat{\mathcal{C}}_B$ to calculate the Floquet multipliers, solutions of Eq. C.8 are computed using one period of data from the RC's response to $\mathcal{C}_A$ and $\mathcal{C}_B$ during the training stage as input. In this case if μ_1, the largest of the resulting μ_i's, is 1 and all other μ_i have magnitude < 1 for both $\mathcal{C}_A$ and $\mathcal{C}_B$ this provides an indication (as these μ_i's are not the Floquet multipliers and are just computed using the same procedure) that the RC's response to the driving input is stable. This is turn provides evidence of whether or not a given configuration of an RC can support the coexistence of both $\mathcal{C}_A$ and $\mathcal{C}_B$ after the training. However, the resulting μ_i's will not verify if the reconstructed attractors satisfy the roundness criteria described previously.

7.2 Parameter sensitivity testing

In this section, the performance of each RC when tasked with solving the extreme case of the seeing double problem is compared and the effects that different training parameters have on multifunctionality are assessed. The training parameters used to generate the results shown in each Figure in this section are provided in Tables 7.1-7.3.

7.2.1 CT-RC: Randomness, ρ, and N

In this section the relationship between changes in the CT-RC's dimension and its ability to achieve multifunctionality are assessed. To be more specific, for 100 random realisations of $\mathbf{M}$ and $\mathbf{W}_{in}$, the regions in the (N, ρ)-plane where the CT-RC exhibits multifunctionality are computed using the roundness analysis procedure outlined in Sec. 7.1.2. Each $\mathbf{M}$ and $\mathbf{W}_{in}$ are constructed according to the method outlined in Sec. 2.3.4. The sparsity and range of element values of these matrices are the same for each random realisation. The reasoning behind this is to gain further insight into the relationship between N, ρ, and multifunctionality while reducing the influence a given random realisation of the respective matrices has on multifunctionality.

Fig	N	P	ρ	σ	γ	β	t_l	t_t
7.1	see Fig	0.05	see Fig	0.2	5	10^{-2}	$6T$	$15T$
7.3	500	0.05	see Fig	0.2	5	10^{-2}	$6T$	$15T$
7.4a	500	0.05	1.4	0.2	see Fig	see Fig	$6T$	$15T$

Table 7.1: CT-RC training parameters in the specified figures. $t_l = t_{\text{listen}}$, $t_t = t_{\text{train}}$, $T =$ period.

Fig	N	P	ρ	σ	η	β	t_l	t_t
7.2	500	0.05	see Fig	0.2	see Fig	10^{-2}	$6T$	$15T$
7.4b	500	0.05	1.4	0.2	see Fig	see Fig	$6T$	$15T$

Table 7.2: LI-RC training parameters in the specified figures. $t_l = t_{\text{listen}}$, $t_t = t_{\text{train}}$, $T =$ period.

Fig	O	k	s	β	t_w	t_t
7.5-7.6	$1, 2$	2	1	see Fig	2	$15T$

Table 7.3: NG-RC training parameters in the specified figures. $t_w = t_{\text{warm}}$, $t_t = t_{\text{train}}$, $T =$ period.

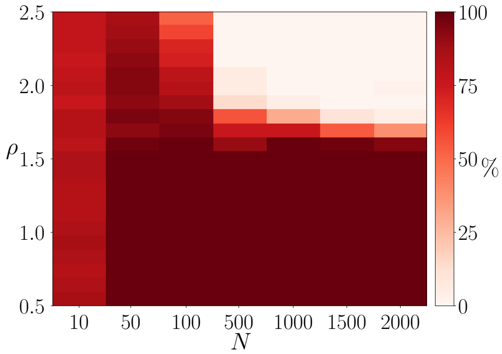
The result of the above experiment for $N \in [10, 2000]$ and $\rho \in [0.5, 2.5]$ is presented in Fig. 7.1 for the case of training the CT-RC to reconstruct only $\mathcal{C}_A$ or $\mathcal{C}_B$, or a coexistence of $\mathcal{C}_A$ and $\mathcal{C}_B$. Each point in the (N, ρ)-plane represents the number of times out of 100 that the CT-RC successfully solved the given task.

In Fig. 7.1a, the result of training the CT-RC to reconstruct only $\mathcal{C}_A$ is illustrated. Here it shows that even when $N = 10$, the CT-RC is capable of reconstructing $\mathcal{C}_A$ for $\rho \in [0.5, 2.5]$ with $> 60\%$ success ratio. When N is increased then an upper bound to ρ appears where for $N \geq 500$ the reconstruction of $\mathcal{C}_A$ is not possible for $\rho > 1.7$. However for $\rho < 1.5$, using $N \geq 500$ the CT-RC achieves reconstruction with $> 90\%$ success ratio. A similar result is also seen in Fig. 7.1b when the CT-RC is trained to only reconstruct $\mathcal{C}_B$. Above all, what Fig. 7.1a and 7.1b show is that a CT-RC with $N = 50$ performs just as well as a network of 2000 neurons for $\rho < 1.5$.

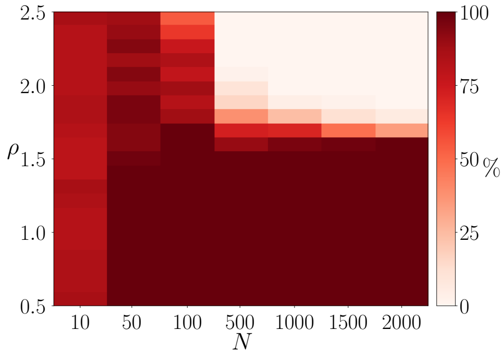
There is a dramatic change in these success rate plots in the (N, ρ)-plane when the CT-RC is trained to reconstruct a coexistence of $\mathcal{C}_A$ and $\mathcal{C}_B$ as shown in Fig. 7.1c. These results show that in order to solve the seeing double problem with $> 90\%$ success then it is only possible in a significantly smaller region of the (N, ρ)-plane in comparison to Figs. 7.1a and 7.1b. Once the CT-RC is tasked with reconstructing a coexistence of $\mathcal{C}_A$ and $\mathcal{C}_B$, there is a clear lower bound in N established whereby for $N = 10$, multifunctionality is not achieved for any of the tested ρ values. Increasing N to 100 provides little improvement with $< 25\%$ of the tested random initialisations of $\mathbf{M}$ and $\mathbf{W}_{in}$ giving rise to multifunctionality. It is only for $N > 1000$ that $> 90\%$ success becomes possible, albeit for only a small range of ρ values.

It is also clear from Fig. 7.1c, that a lower bound to the ρ values emerges as it is only for $\rho > 1$ that multifunctionality is achieved regardless of N. In addition, the upper bound to ρ from Figs. 7.1a and 7.1b remains a feature of the (N, ρ)-plane in Fig. 7.1c as multifunctionality is not achieved for $\rho > 1.7$.

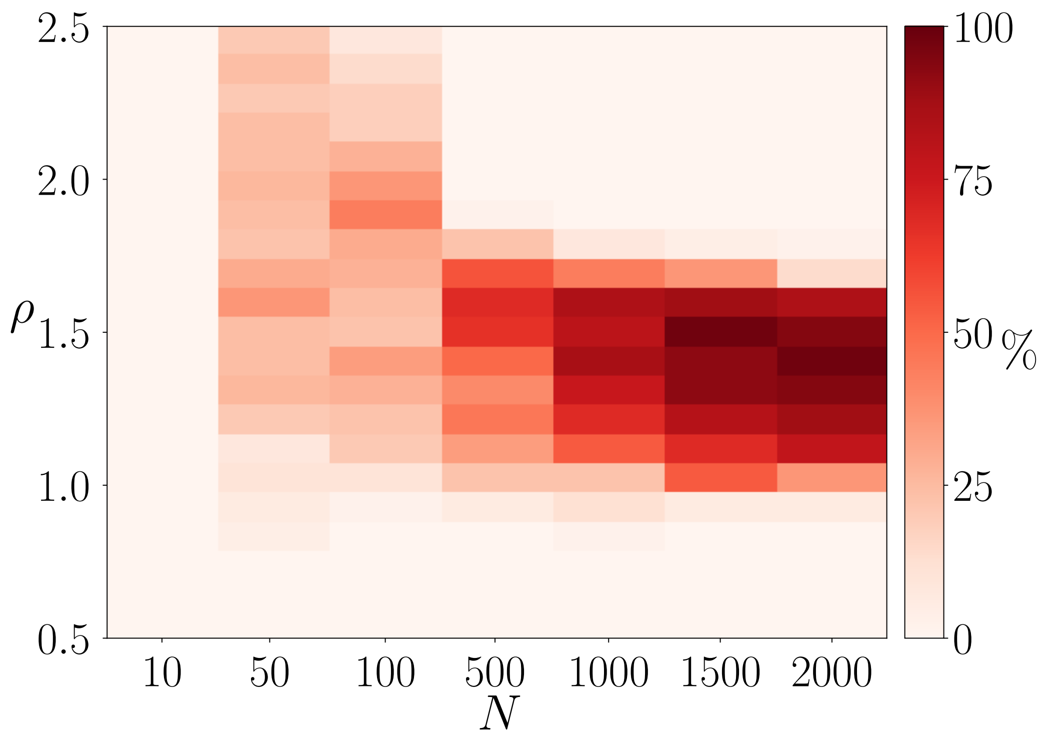
Fig. 7.1c is also in good agreement with the results shown in Fig. 6.4 as the $N = 1000$ CT-RC studied in the previous Chapter was shown to achieve multifunctionality in the case of $x_{cen} = 0$ for $\rho \in [1.1, 1.4]$.



(a) $\mathcal{C}_A$



(b) $\mathcal{C}_B$



(c) $\mathcal{C}_A$ and $\mathcal{C}_B$

Figure 7.1: Success rate in the (N, ρ)-plane of 100 random realisations of the CT-RC when trained to reconstruct either $\mathcal{C}_A$ in (a), $\mathcal{C}_B$ in (b), or a coexistence of $\mathcal{C}_A$ and $\mathcal{C}_B$ in (c).

Overall, the results presented in Fig. 7.1, highlight the important role that N and ρ play in whether a given CT-RC achieves multifunctionality, however, many questions remain. The motivation behind the experiments carried out in the following sections are two-fold. The first is to determine if the STM metric or Floquet analysis procedure outlined in Sec. 7.1.2 are capable of identifying whether a given random realisation of a LI and CT-RC are capable of achieving multifunctionality. The second is to explore whether different parameter settings of these RC's can provide improved performance at smaller N values.

7.2.2 LI-RC: Randomness, ρ, η, and memory capacity

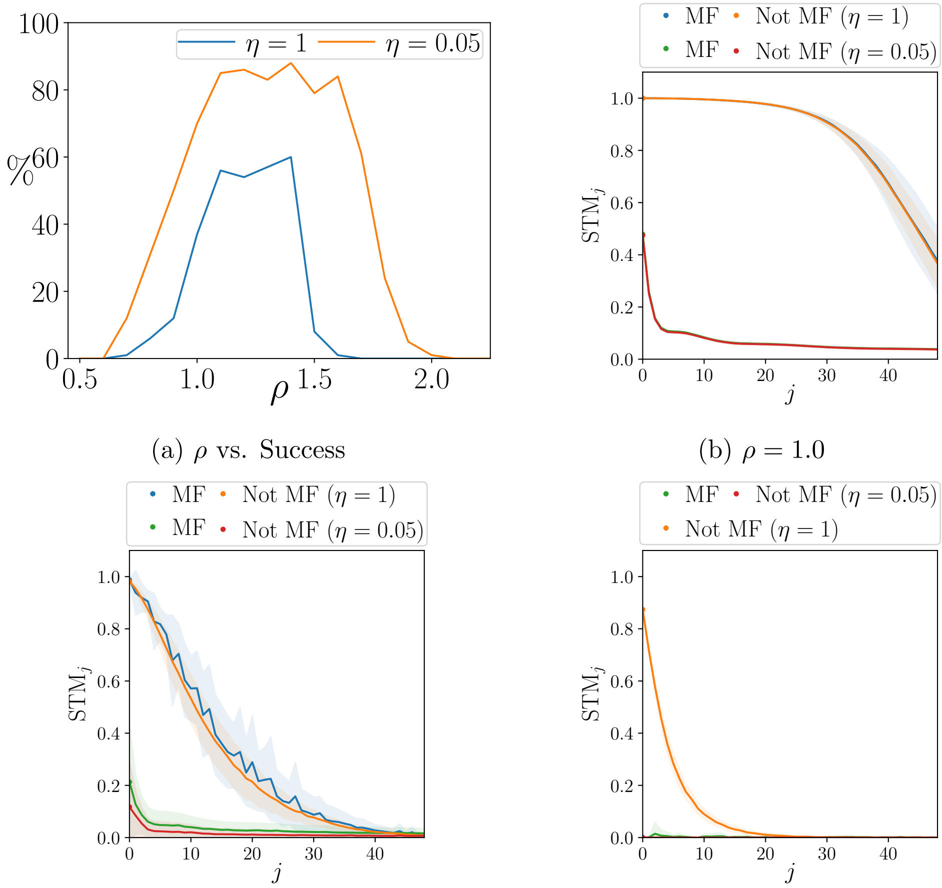
In this section, the performance of the LI-RC setup with $N = 500$ is examined at different ρ and η values to determine if the STM metric is successful at distinguishing between whether different random realisations of the LI-RC can give rise to multifunctionality.

Fig. 7.2a illustrates the number of times out of 100 random realisations of $\mathbf{M}$ and $\mathbf{W}_{in}$ that a given LI-RC achieves multifunctionality for different values of ρ as determined by the roundness analysis procedure described in Sec. 7.1.2. For $\eta = 0.05$ it is shown here that these LI-RCs outperform the LI-RCs with $\eta = 1$ resulting in not only a broader range of ρ values where multifunctionality was achieved but also a greater probability of success.

The STM metric is used here to assess if, in this sense, a given LI-RC's 'memory' is critical to achieving multifunctionality. The rationale behind this metric and how to compute it is outlined in Sec. 7.1.2.

The result of computing the STM for 10 LI-RCs which are known to achieve multifunctionality and for 10 LI-RCs that do not solve the seeing double problem, as chosen from the results shown in Fig. 7.2a, are illustrated in Figs. 7.2b-7.2d for $\eta = 0.05$ and $\eta = 1$ at different ρ values. What is shown here is the relationship between j and STM_j, the STM computed for a given j, and the standard deviation of STM_j across the different random realisations of the LI-RC.

What is common across Figs. 7.2b-7.2d is that there is no significant differ-



(a) ρ vs. Success

(b) $\rho = 1.0$

(c) $\rho = 1.4$

(d) $\rho = 1.8$

Figure 7.2: LI-RC: (a) Success rate vs. ρ of 100 random realisations of the LI-RC when trained to solve the seeing double problem. (b)-(d) STM of RC's that fail and succeed to solve the seeing double problem with error bars given by the standard deviation. j is the time shift used in calculating the STM. MF describes when multifunctionality was achieved.

ence $\forall$ the computed STM_j's amongst successful or unsuccessful realisations of the LI-RCs. This shows that there is no correlation between whether a given LI-RC achieves multifunctionality and its STM.

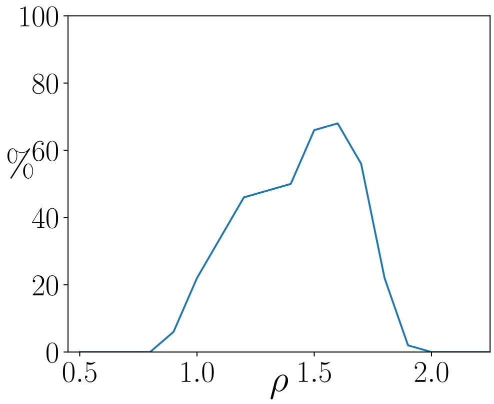
Furthermore, there is a significant difference in the evolution of STM_j as j increases for the $\eta = 1$ and $\eta = 0.05$ curves for the ρ values studied in Figs. 7.2b-7.2d. At small j the corresponding STM_j for $\eta = 0.05$ tends to 0 relatively quickly, it is only for large j that the STM_j for $\eta = 1$ goes to 0. For both $\eta = 1$ and 0.05, the rate of convergence of STM_j to 0 decreases as ρ increases. Given that the LI-RC performed much better for $\eta = 0.05$ as shown in Fig. 7.2a, the above discussion provides further clarification that the STM metric gives no indication on whether a given RC has the capacity to perform well on the seeing double problem or achieve multifunctionality.

7.2.3 CT-RC: Randomness, ρ and Floquet analysis

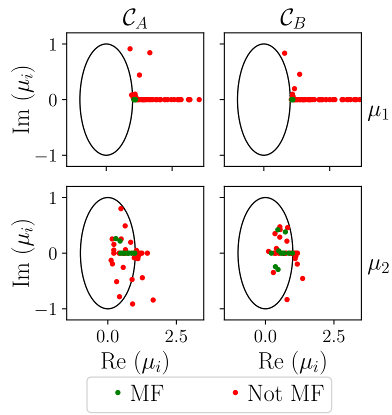
In this section, the performance of the CT-RC setup with $N = 500$ is examined at different ρ values to determine if the Floquet-type analysis procedure outlined in Sec. 7.1.2 is successful at distinguishing between whether different random realisations of the CT-RC can give rise to multifunctionality.

The same information from Fig. 7.1 for $N = 500$ is illustrated from a different perspective in Fig. 7.3a to more easily compare the performance of the CT and LI-RCs for changes in ρ. In contrast to the LI-RCs with $\eta = 0.05$, these CT-RCs with $\gamma = 5$ are able to achieve multifunctionality with a success ratio $> 50\%$ for a smaller range of ρ values but still performs much better than the LI-RCs with $\eta = 1$ as indicated by the area under these curves.

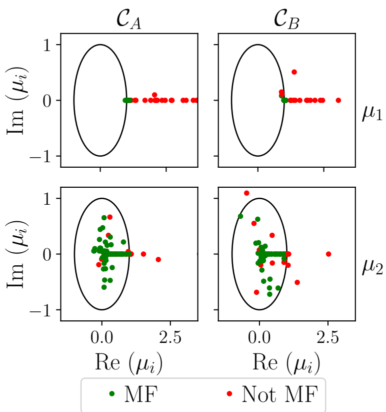
Using the information which Fig. 7.3a provides on whether a given CT-RC achieves multifunctionality, Figs. 7.3b-7.3d show that there is a direct correlation between CT-RCs which solve the seeing double problem and the results of the corresponding Floquet analysis at different values of ρ.



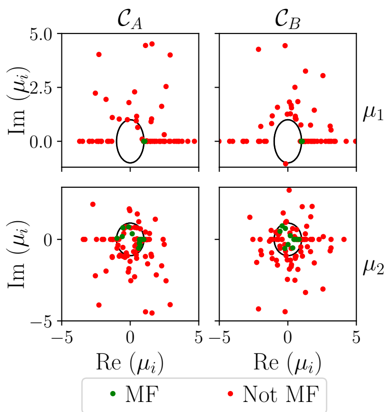
(a) ρ vs. Success



(b) $\rho = 1.0$

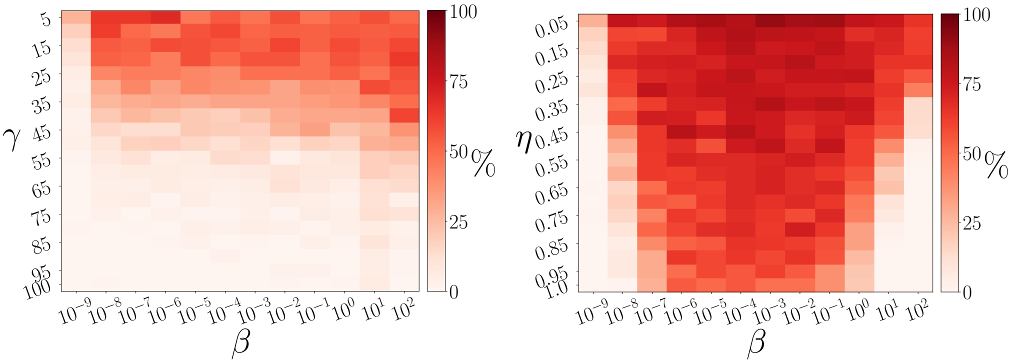


(c) $\rho = 1.4$



(d) $\rho = 1.8$

Figure 7.3: CT-RC: (a) Success rate vs. ρ of 100 random realisations of the CT-RC when trained to solve the seeing double problem. (b)-(d) Corresponding two largest μ_i's, μ_1 and μ_2, for both $\mathcal{C}_A$ and $\mathcal{C}_B$ for these $\mathbf{M}$ and $\mathbf{W}_{in}$ at specified ρ. MF describes when multifunctionality was achieved.



(a) CT-RC (b) LI-RC

Figure 7.4: Success rate in the (β, γ)-plane and the (β, η)-plane for 100 random realisations of the CT and LI-RCs.

7.2.4 CT & LI RCs: Randomness, γ, η, and β

Based on the results presented in Fig. 7.1c, the aim of the present section is to determine whether the performance of the LI and CT-RCs with $N = 500$ and $\rho = 1.4$ is improved at different values of η, γ, and β.

Fig. 7.4 illustrates the number of times out of 100 that a given pair of (β, η) or (β, γ) values give rise to multifunctionality using the roundness criteria as described in Sec. 7.1.2.

Fig. 7.4a shows no significant improvement in the performance of the CT-RC for different β and γ values. In particular, for $\gamma > 40$ the majority of random realisations of the CT-RCs do not solve the seeing double problem. Whereas for the LI-RC, in Fig. 7.4b, its best performance occurs for the same $\eta = 0.05$ studied in Sec. 7.2.2 with 80% success rate for $\beta \in (10^{-8}, 10^{1})$.

Furthermore, Figs. 7.4a and 7.4b show that the limit of $\eta = \gamma\tau$ for $\tau \to 0$, which relates the CT-RC to the LI-RC as discussed earlier in Sec. 2.3.5, holds for only small values of η and γ.

Overall, Fig. 7.4 shows that there is a much wider range of η and β values where over 70% of the tested LI-RCs achieve multifunctionality in comparison to the corresponding γ and β values tested across the different random realisations of the CT-RC. What is common between both Figs. 7.4a and 7.4b is the role of β. If β is too small then both RCs have a success rate $< 10\%$

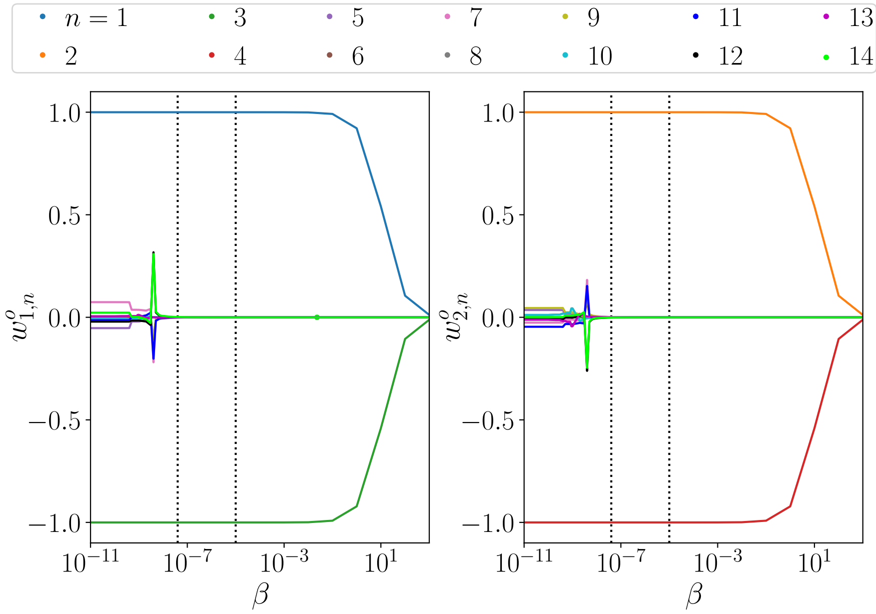


Figure 7.5: β vs. $w^o_{m,n}$, the elements of $\mathbf{W}_{out}$ for the NG-RC setup. The region enclosed by the vertical dotted lines highlights the β values where multifunctionality was achieved.

and while larger β values prevent overfitting, if β is too large this also inhibits the RCs from learning the correct dynamics.

7.2.5 NG-RC: β and $\mathbf{W}_{out}$

The performance of the NG-RC is now assessed when tasked with solving the seeing double task with parameters specified in Table 7.3. Note the square readout function, $\mathbf{q}(\cdot)$, is not used to obtain $\mathbf{W}_{out}$. As a result, the trained $\mathbf{W}_{out}$ is $\in \mathbb{R}^{2\times 14}$. The relationship between each element of this $\mathbf{W}_{out}$, denoted as $w^o_{m,n}$, and β is illustrated in Fig. 7.5.

The vertical dotted lines in Fig. 7.5 indicate the range of β values where the NG-RC reconstructs both $\mathcal{C}_A$ and $\mathcal{C}_B$. For $\beta < 4 \times 10^{-8}$, the state of the NG-RC tends to infinity, and for $\beta > 10^{-5}$ the system tends to the fixed point at the origin.

Fig. 7.5 shows that when multifunctionality is achieved there are four $\mathbf{W}_{out}$

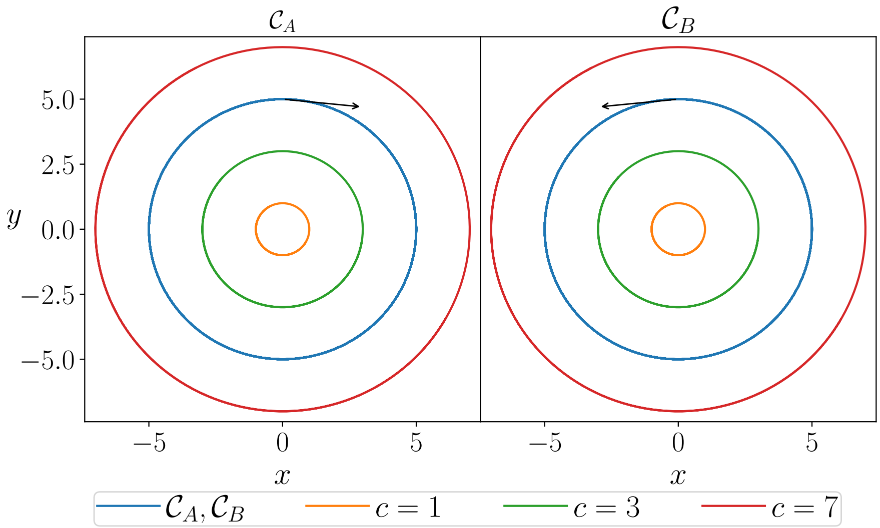


Figure 7.6: Examples of the trained NG-RC's output dynamics for different choices of initial conditions.

elements, $w^o_{1,1} = 0.99989995, w^o_{1,3} = -0.99999995, w^o_{2,2} = 0.99990005$, and $w^o_{2,4} = -1.00000005$ which are $\not\approx 0$. Given the architecture of the NG-RC it is possible to write the trained system as,

$$x(t+1) = x(t) + \Delta x(t) = (1 + w^o_{1,1})x(t) + w^o_{1,3}x(t-1),$$
$$y(t+1) = y(t) + \Delta y(t) = (1 + w^o_{2,2})y(t) + w^o_{2,4}y(t-1).$$

The resulting governing equations of the NG-RC are uncoupled and linear, like the driving input described in Eq. 6.2. As a consequence, the NG-RC has not reconstructed limit cycles but has produced a set of equations that can mimic the input training data. As a further consequence, Fig. 7.6 shows that by initialising the trained NG-RC with two consecutive points on a circle of any given radius it produces the corresponding circular trajectories.

It is also important to note that despite, $w^o_{1,1} \approx 1, w^o_{1,3} \approx -1, w^o_{2,2} \approx 1$, and $w^o_{2,4} \approx -1$, the NG-RC becomes unstable if the values 1 and -1 are used for the weights.

7.3 Conclusion

The results presented in Sec. 7.2 highlight that the LI-RC is the most capable of the tested RCs to solve the seeing double problem for the case of $x_{cen} = 0$. At the same time, the LI-RC has an advantage over the CT-RC at solving tasks which involve overlapping training data as the state of the LI-RC is able to intersect itself whereas the CT-RC's state cannot.

Despite stipulating that memory is a key element for the RCs to achieve multifunctionality in the case of overlapping training data, Figs. 7.2b-7.2d demonstrate the shortcomings of a commonly used 'short term memory' metric as it cannot be used to distinguish between whether a given random realisation of the LI-RC can give rise to multifunctionality. Instead, Figs. 7.3b-7.3d show that the Floquet analysis procedure outlined in Sec. 7.1.2 provides a more rigorous assessment.

At the same time, the results on memory shown in Sec. 7.2.2 subsequently motivates the need to design a metric which sufficiently describes the degree to which a given RC can achieve multifunctionality. While this goes beyond the scope of this Thesis, from the results presented in this Chapter and previous Chapters, it is clear that such a measure would need to take into account not only the amount of attractors a given RC can reconstruct a coexistence of but also the relationship between these attractors like the timescale of each attractors dynamics and whether the training data is overlapping.

There are many other avenues of exploration in terms of following on from the results presented in this Chapter. Aside from assessing additional parameter relationships in the CT and LI-RC designs such as tuning the input strength, σ, or changing the networks sparsity, P, the major hurdle to overcome, in particular for the CT-RC, is to decrease the dimension of the RC in order to reduce the computational cost but without compromising on the success rate of high dimensional RCs like those shown in Fig. 7.1c. Additional motivation towards moving beyond straightforward parameter sensitivity tests comes from Fig. 7.4 which shows no major improvements to the $N = 500$ CT-RC for changes in γ and β. A more fortuitous approach may be to apply the methods used in [150] where Carroll and Hart demonstrate that by training a model of an opto-electronic delay-based RC using time-shifted copies of its outputs significantly improves the performance of low-dimensional versions of this RC on tasks involving model-free inference of unmeasured variables.

Furthermore, in terms of pushing back the lower bound of ρ values where multifunctionality is first achieved in Fig. 7.1c, progress may also be made through reconsidering how the RC 'sees' the seeing double problem by changing the design of $\mathbf{W}_{in}$. For instance, the influence of each row in $\mathbf{W}_{in}$ only having one randomly chosen nonzero element is clear from Figs. 6.12 and 6.13 given the diagonal nature of the RC's solutions until multifunctionality is achieved at sufficiently large ρ values. Therefore by choosing a different initialisation scheme for $\mathbf{W}_{in}$ which incorporates the circular nature of the training data, such as using the Marsgalia polar method to generate random numbers on a disk, this may decrease the role these untrained dynamics play and allow the RC to solve the seeing double problem at smaller ρ.

The differences between how each RC solves the seeing double problem also comes to light in this Chapter. In order for the CT-RC and LI-RC to achieve multifunctionality in this scenario, these RCs are required to convert this overlapping driving input data into attractors with two distinct basins of attraction that can be projected to resemble the desired coexistence. While in Sec. 7.2.5, the NG-RC is found to solve the seeing double problem in a different way due to its design. Despite configuring the NG-RC with quadratic terms in its design, the trained NG-RC is an uncoupled set of linear equations. As a consequence, instead of generating trajectories on limit cycles, this linear system produces centers of any given radius as shown in Fig. 7.6.

Most importantly, the results presented in this Chapter establish a guideline for choosing RC design and training parameters not only for solving the seeing double problem but as this task encapsulates the issues of overlap in the hardest but simplest of scenarios, these results provide a template for training RCs to achieve multifunctionality in various circumstances.

Chapter 8

Modelling Seizure Dynamics

The results presented in this Chapter demonstrate how multifunctionality can be exploited as a tool to produce data-driven models of physical phenomena. By translating the behaviour of epileptic neurons to the language of dynamical systems and multifunctionality, this Chapter demonstrates how to train a RC to reconstruct a coexistence of the seizure and non-seizure states present in experimental data. It is found that by destabilising the multifunctional RC to exhibit chaotic itinerancy the dynamical features of epilepsy can be modelled.

8.1 Multifunctionality as a modelling tool

Deciphering the inner workings of the brain remains one of the most challenging problems in science which exhausts and extends much of our current knowledge of dynamical systems [106, 151]. Fortunately, mathematical descriptions of neurological conditions have led towards improved modelling of certain 'dynamic diseases' [152]. Furthermore, with the increasing advancements in ML, training ANNs to model phenomena from experimental data provides researchers in neuroscience with a range of new tools to generate and test further neural hypotheses [126].

In this Chapter, the above rationale is employed in order to model dynamical transitions present in experimental data obtained from epileptic rodent brains using a multifunctional RC. This work establishes the first ML application of multifunctionality as a modelling tool.

8.1.1 Obtaining the data[1]

Electrophysiological data were acquired from male *Genetic Absence Epilepsy Rats from Strasbourg* (GAERS) aged between 4-7 months. This involved using silicon-site electrodes to sample voltages from the *ventrobasal thalamus* while the rats were able to move freely and alternate between waking, sleeping, and spontaneously seizing states. This data was processed with a Plexon HST/32V-G20 VLSI-based preamplifier and associated digitization system at 20000 samples/second and subsequently down sampled to 1000 samples/second. For more information see [153].

An example of the voltage activity patterns obtained from the above experiment is shown in Fig. 8.1. This plot illustrates the recordings of a number of measurement channels, labelled as 'K1M', 'T8C', and 'K3A', which are taken from different GAERS nearby a transition from non-seizure activity to seizure activity. A small amplitude slow oscillation about $0\,\mathrm{mV}$ (i.e., deviations from long term mean of $0.1\,\mu\mathrm{V}$) is a distinctive feature of the non-seizure activity while the seizure activity patterns oscillate with a much larger amplitude on a much faster timescale. These GAERS are not subject to external stimuli which directly trigger these transitions, instead this switching behaviour occur spontaneously and is a natural feature of the GAERS neural activity.

8.1.2 Dynamical interpretation of the data

The data shown in Fig. 8.1 is now reinterpreted from a dynamical systems perspective in order make use of the tools developed Sec. 3.2.2 to produce a data-driven model which incorporates multifunctionality to simulate the neural activity patterns from these GAERS.

[1]The experimental data discussed here was obtained by Cian McCafferty, François David, and Vincenzo Crunelli, all of whom are co-authors on a paper with the author of this Thesis, Vassilios A. Tsachouridis, and Andreas Amann. This paper is titled, 'Modelling dynamical features of epilepsy with a multifunctional reservoir computer' and is still in development at the time this Thesis was written. The details on the experimental data discussed in this section was written by Cian McCafferty.

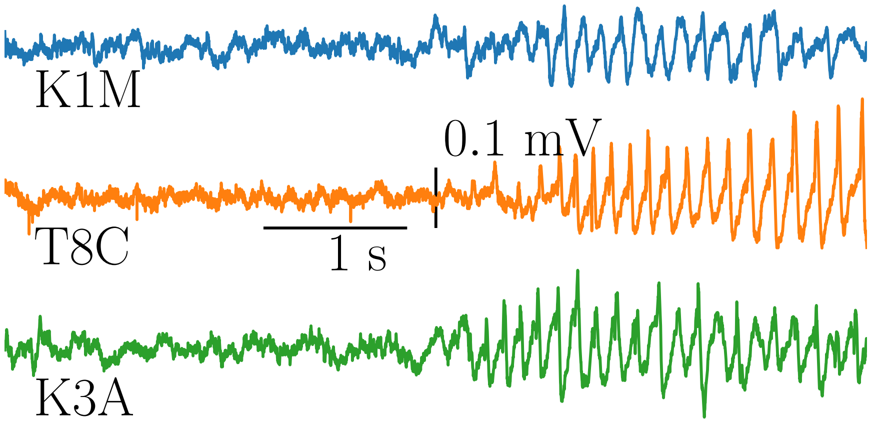


Figure 8.1: Example of the neural voltage data near the transition between non-seizure and seizure activity taken from the measurement channels, K1M, T8C, K3A with the timescale and amplitude of oscillation as indicated.

Seizure and non-seizure attractors

The 'K1M', 'T8C', and 'K3A' measurement channels shown in Fig. 8.1 are now rewritten as variables which are functions of time, $\mathrm{T8C}(t), \mathrm{K3A}(t), \mathrm{K1M}(t)$, that describe the evolution of the respective measurement channels in time. The minimum of the single longest time spent in seizure from the three variables is determined in order to form the time series data set $\mathbf{Y}_{\mathcal{P}_1} = [\mathbf{u}_{\mathcal{P}_1}(t_m) \; \ldots \; \mathbf{u}_{\mathcal{P}_1}(t_n)]$ where this minimum time was found to be $90\,s$, $\mathbf{u}(t) = (\mathrm{T8C}(t), \mathrm{K3A}(t), \mathrm{K1M}(t))$, and these measurements are down sampled at every $\tau = 0.001\,s$ and the resulting discrete-time interval is defined from $t_m = 1$ to $t_n = 90001$. $\mathbf{Y}_{\mathcal{P}_1}$ is considered to represent a sufficient amount of information of a trajectory on a 'seizure attractor', $\mathcal{P}_1$. The same procedure is used to select neural data representing a trajectory on the 'non-seizure attractor', $\mathcal{P}_2$, for the same length of discrete-time interval, $[t_m, t_n]$.

By interpreting the non-seizure and seizure dynamical regimes as trajectories on different attractors, it becomes possible to train a RC to reconstruct a coexistence of these attractors. However, the key dynamical feature of the neural data is not the coexistence of these attractors but rather the switching dynamics between these attractors. There are many dynamically plausible mechanisms which can in principle describe these transition and examples of this are discussed later in Sec. 8.2.

Nevertheless, in order to accurately model the transitions observed in these GAERS, this requires the resulting model to produce transitions between seizure and non-seizure activity without using an external stimulus. At the same time, Fig. 5.6 shows that a RC can produce this behaviour as, in this example, when the RC loses its ability to exhibit multifunctionality it enters into the dynamical regime of chaotic itinerancy. For further information on chaotic itinerancy, and how multifunctionality presents itself as a suitable means to realise it, see the discussion in Sec. 3.4.4. Furthermore, while chaotic itinerancy serves as a useful means to generate the transitions between seizure and non-seizure behaviour, the results presented in this Chapter are by no means intended to insinuate that chaotic itinerancy plays a role in the dynamics of the epileptic brain.

The results presented in Sec. 8.3 demonstrate how the dynamics of a multifunctional RC can be destabilised to produce chaotic itinerancy and thereby bring about transitions between ruins of the reconstructed seizure and non-seizure attractors.

Generating the training data

The voltage time series data sets, $\mathbf{Y}_{\mathcal{P}_1}$ and $\mathbf{Y}_{\mathcal{P}_2}$, are used drive a response from the continuous-time RCs described in Eqs. (3.7) and (3.9) in order to generate the corresponding response data matrices, $\mathbf{X}_{\mathcal{S}_1}$ and $\mathbf{X}_{\mathcal{S}_2}$, to compute $\mathbf{W}_{out}$ according to the blending technique outlined in Sec. 3.2.2 for a given blending parameter, α. In this instance, $\mathcal{S}_1$ and $\mathcal{S}_2$, describe the RC's response to the seizure and non-seizure driving inputs.

Referring back to the discussion in Sec. 3.2.1, the role of the each transformation function, $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$, in the present task is to lift $\mathcal{P}_1$ and $\mathcal{P}_2$ to attractors $\mathcal{S}_1$ and $\mathcal{S}_2$ that are able to coexist in the RC's state space, $\mathbb{S}$, after closing the loop with $\mathbf{W}_{out}$. Multifunctionality is achieved when this $\mathbf{W}_{out}$ is able to project trajectories on $\mathcal{S}_1$ and $\mathcal{S}_2$ to resemble $\mathcal{P}_1$ and $\mathcal{P}_2$ in the prediction state space, $\mathbb{P}$.

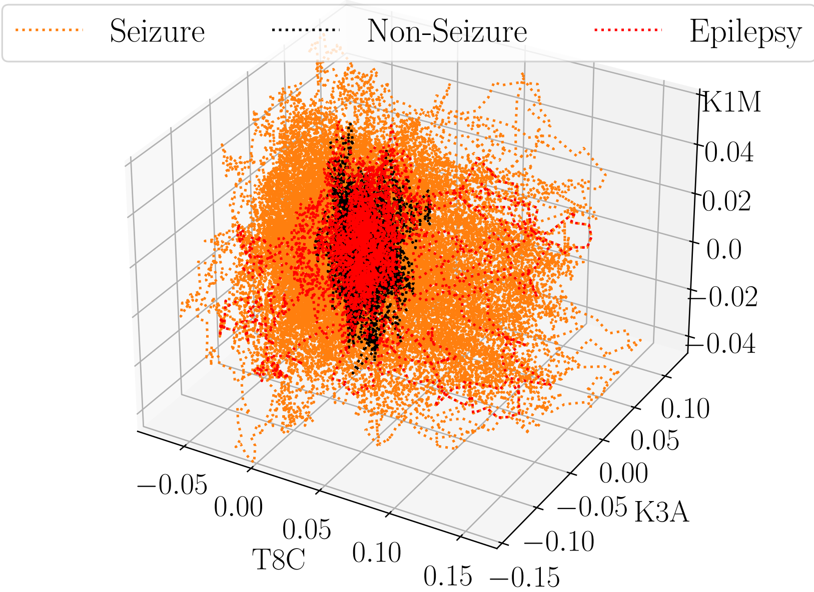


Figure 8.2: Non-seizure (in black) and seizure (in orange) neural dynamical regimes of the experimental data in a three dimensional space described by the voltage measurement channels. The transition between non-seizure to seizure and back to non-seizure is plotted in red.

Why use multifunctionality?

In Fig. 8.2, $\mathbf{Y}_{\mathcal{P}_2}$ and $\mathbf{Y}_{\mathcal{P}_1}$ are plotted in black and orange respectively in the corresponding T8C-K3A-K1M space. When the data is viewed from this perspective, the motivation behind using multifunctionality in this data-driven modelling approach becomes clear as the 'attractors', $\mathcal{P}_1$ and $\mathcal{P}_2$, appear as if they are in coexistence with one another.

The data plotted in red in Fig. 8.2 depicts what the transition between the non-seizure, seizure, and back to the non-seizure regimes looks like in this state space. This red data is labelled as 'Epilepsy' to indicate the difference between the behaviour of a healthy brain, whose activity is confined within the non-seizure data, and an unhealthy brain, whose activity patterns switch between seizure and non-seizure activity.

In an analogous sense, the dynamics of a multifunctional RC resembles the behaviour of a healthy brain whereby there are no transitions between the coexisting attractors. However, there remains a possibility that a transition can indeed occur if the system receives input from an external stimulus which can force the state of the multifunctional RC away from the non-seizure attractor to the seizure attractor.

8.2 Existing dynamical models of epilepsy

There are many existing dynamical systems models which focus on describing the transitions between non-seizure and seizure neural activity from the perspective of multistablity.

In [154–157] the first multistable dynamical systems model of transitions in the epileptic brain was studied. In [154] and [155], different mechanisms are put forward which describe how transitions between a 'non-seizure' and 'seizure' attractor (with variables described in terms of voltage activity patterns from neurons) can occur from a biological perspective. These results stem from a biological data informed model presented in [156], and further explored in [157], which are an extension of the transfer function based model studied in [158].

The model in [156] and [157] is a 'lumped model' meaning that the explicit behaviour of individual neurons is not simulated and instead the 'mean membrane potential' (voltage) of a particular type of neuron (pyramidal) from populations of interacting neurons (pyramidal, interneurons, thalamocortical, and reticular thalamic), integrating neuronal and network properties into the model simulated in Simulink. The bifurcation analysis presented in [156] shows that by varying the 'cortical input' parameter, a hysteresis cycle in the system's mean membrane potential between a fixed point (describing the non-seizure attractor) and a limit cycle (describing the seizure attractor), is found to occur as a result of a subcritical Hopf bifurcation. Transitions between non-seizure and seizure attractors in the bistable region mentioned above are seen to occur by introducing noise into this model. The distribution between different durations of time spent on each attractor are recorded and compared to electroencephalogram (EEG) data from epileptic 'Wistar albino Glaxo from Rijswijk' (WAG/Rij) rats and are shown to have qualitatively similar exponential decay distributions, spending more time on the non-seizure attractor.

In [157] the above study is extended to incorporate data from GAERS. Counter-stimuli methods are also investigated in [156] and [157] in order to control the occurrence of seizures and these are shown to perform effectively. The authors highlight the scope and limitations of their model in [156] and [157], and pose interesting questions on whether their model can account for seizures in humans, be used to predict the onset/existence of seizures, and study the effects of antiepileptic drugs.

The connection between epilepsy and multistable dynamics was also explored by Breakspear *et al.* in [159] using a mean field neural model presented in [160]. Similar to the previous example, a hysteresis cycle is also found by varying a cortical input parameter and the transitions between seizure and non-seizure attractors are induced by varying this parameter as the model evolves in time. The behaviour of this model also aligns with EEG recordings from epileptic humans.

A multistable network model was presented in [161] which is designed with bidirectional nearest neighbour synaptic coupling between neurons and the dynamics of the individual neurons are governed by the 4-dimensional 'neocortical model' developed in [162] which is derived from the Hodgkin-Huxley model [163]. In [161] the researchers investigate the dynamics of coupled neurons in the case of 2 and 20 neurons. An independent Gaussian white noise term is added to each membrane potential evolution equation in the network which gives rise to noise induced transitions between the coexisting attractors.

In [164] the dynamics of a 10-dimensional model was investigated. This model is based on an 8-dimensional model introduced in [165] and is considered as an extended version of the previously mentioned model presented in [159]. The model in [164] is capable of producing different types of EEG-like dynamics over much wider regions of parameter space than the delay-differential equation (DDE) model in [165]. In particular, a broad region of bistability between a non-seizure (fixed point) and seizure (spike-and-wave limit cycle) attractor is found in [164] which is more representative of the clinical recordings shown in the paper.

The 'two compartment model' was introduced in [166]. This model consists of 8 ODEs that describe the behaviour of a given population of neurons and these populations are coupled via a weighted contribution of excitatory input from all other populations. This model is derived from the cortical neural mass activity model presented in [167, 168]. A bifurcation analysis of a single node, two node, and 25 node network with homogeneous coupling reveals broad regions of parameter space where a bistability between non-seizure and seizure attractors are found. The greatest difference in the dynamics of this model in comparison to those mentioned previously is that instead of the non-seizure attractor being described by a fixed-point (in the case where there is no external noise added to the system) it is a low-amplitude high-frequency period-1 limit-cycle. The seizure attractor is a limit-cycle with a much larger amplitude and smaller frequency of oscillation with multiple harmonics, which is similar to the previous examples. By adding noise to the system, transitions between these attractors occur and the researchers investigate the nature of these transitions through the lens of intermittency. Following the intermittency analysis of EEG signals in [169, 170], the researchers examine the distribution of a given duration of time spent on the non-seizure attractor over many transitions and find that this form of intermittency is not consistent with the power law distribution indicative of on-off intermittency (as found in [170]) or those which follow indications of type-III intermittency (inverse period-doubling bifurcation) as found in [169]. Instead this form of intermittency is more in agreement with intermittencies associated with a global bifurcation. The authors admit that they cannot confirm this as their model may allow for different types of intermittent solutions which have not yet been rigorously characterised.

In order to shed some light on the above and explore the dynamic mechanisms underlying this form of intermittency, in [171] Goodfellow and Glendinning investigated the dynamics of a low dimensional model which preserves fundamental properties of the system studied in [166]. Bistability remained a key dynamical property of this reduced model and intermittency was produced through additive homogeneous coupling in a 3 neuron network. Over a long simulation of the model, where many transitions occurred, a U-shaped distribution of the duration of time spent on the non-seizure attractor was found which is indicative of type-I intermittency (saddle-node bifurcation).

There have also been a number of other investigations which assess EEG data for evidence of intermittency as a mechanism to describe the transition between non-seizure and seizure attractors in the epileptic brain [172, 173].

8.3 Modelling dynamical features of epilepsy

In this section the results of training the RC to reconstruct a coexistence of seizure and non-seizure attractors are presented. Following this, a route to generating chaotic itinerancy with this multifunctional RC is identified. The resulting RC's state wanders between ruins of the seizure and non-seizure attractors thereby resembling the signature dynamical features of the transitions present in the experimental data.

8.3.1 Preparing the data and RC

Prior to training the RC, the voltage time series data sets, $\mathbf{Y}_{\mathcal{P}_1}$ and $\mathbf{Y}_{\mathcal{P}_2}$ are first filtered using the 'savgol_filter' function from the scipy.signal library in Python. This is the Savitzky-Golay filter which fits a polynomial of a given order to successive intervals of data points. In the case of the non-seizure and seizure data, window lengths of 201 and 51 data points are taken and polynomials of order 5 are fitted to these data points. This step smoothens the training data and was found to improve the RC's ability to reconstruct a coexistence of both attractors.

To highlight the differences between the experimental data and the filtered data used in the training of the RC, the original and filtered versions of $\mathbf{Y}_{\mathcal{P}_1}$ and $\mathbf{Y}_{\mathcal{P}_2}$ are illustrated in Fig. 8.3. Here it can be seen that certain aspects of the original neural data are lost, most notably, the filtered data occupies

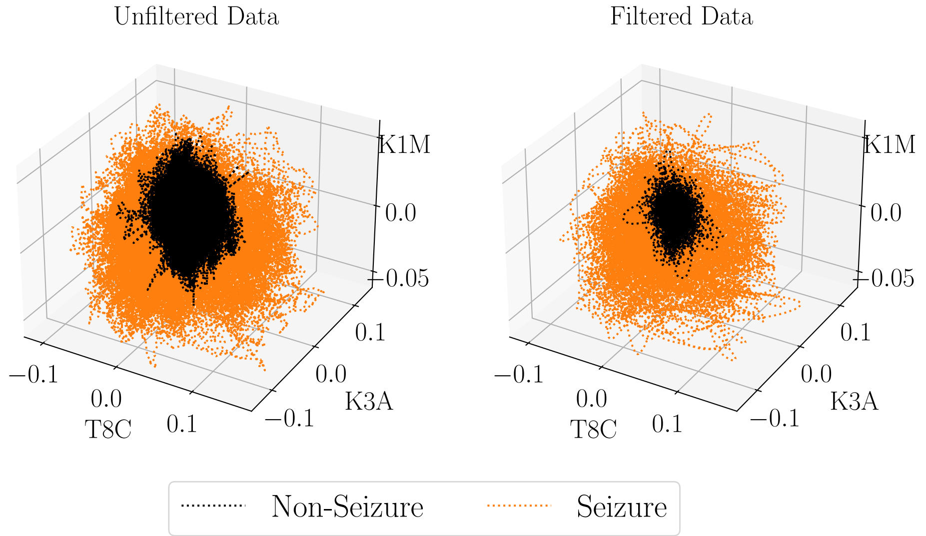


Figure 8.3: Unfiltered experimental data vs filtered data used to train the RC.

a slightly smaller volume of the state space. Despite eroding some of these features, the main characteristics of the non-seizure attractor resembling a slow oscillation about 0 mV and the much faster and larger oscillation of the seizure attractor remain.

At the same time, filtering the data is not totally unreasonable as Fig. 5.6 shows that prior to a given system exhibiting chaotic itinerancy the respective attractors occupy an increasingly larger volume of state space.

However, there are two further traits of the training data which are known to inhibit RCs from achieving multifunctionality, the large amount of overlap and the different timescales between the seizure and non-seizure attractors. The issue of overlapping training is explored at length in Chapters 6 and 7, Fig. 4.9 demonstrates the difficulties posed for the RC when tasked with reconstructing chaotic attractors with increasingly different timescales. Be that as it may, since the training data has already lost some of its original attributes from the filtering process, by reducing or removing the overlap between the data or tweaking the timescales to match one another further dilutes the connection between the experimental and training data.

Taking the above into account, an alternative approach to mitigate these

N	P	ρ	σ	γ_{min}	γ_{max}	Γ	β	α	t_l	t_t
1000	0.1	1.35	5	$10^{-0.4}$	$10^{0.12}$	135	200	0.1025	10	50

Table 8.1: Hyperparameters and time parameters used to generate the multifunctional RC. Note, $t_l = t_{\text{listen}}$ and $t_t = t_{\text{train}}$.

issues is to adapt the RC from its design in Eq. (2.10) to include a term that infuses more diverse timescales into the RC's dynamics like in the method introduced in [174]. This adapted RC is described according to,

$$\dot{\boldsymbol{r}}(t) = \boldsymbol{\gamma}\left[-\boldsymbol{r}(t) + \tanh\left(\mathbf{M}\,\boldsymbol{r}(t) + \sigma\mathbf{W}_{in}\,\boldsymbol{u}(t)\right)\right], \tag{8.1}$$

$$\boldsymbol{r}(0) = \mathbf{0}^T, \tag{8.2}$$

where the elements of $\boldsymbol{\gamma} \in \mathbb{R}^N$ are drawn from a log-uniform distribution with γ_{min} and γ_{max} chosen to reflect the training data in order to introduce further multiscale dynamics in the RC. The resultant log-uniform distribution is multiplied by a scalar, Γ. From preliminary tests, this technique is found to improve the performance of the RC without making further compromises in preparing the training data for the RC.

$\mathbf{M}$ and $\mathbf{W}_{in}$ in Eq. (8.1) are constructed according to the method outlined in Sec. 2.3.4. To train the listening RC in Eq. (8.1) to become multifunctional and reconstruct a coexistence of $\mathcal{P}_1$ and $\mathcal{P}_2$, the blending technique outlined in Sec. 3.2.2 is used. Solutions of Eq. (8.1) in response to the driving inputs from $\mathcal{P}_1$ and $\mathcal{P}_2$ are computed using the 4^{th} order Runge-Kutta method with integration time-step, $\tau = 0.001$, which is the same as the measurement interval of the time-series data that describes $\mathcal{P}_1$ and $\mathcal{P}_2$. The RC training parameters used to reconstruct a coexistence of $\mathcal{P}_1$ and $\mathcal{P}_2$ are specified in Table 8.1.

8.3.2 Reconstructing a coexistence of $\mathcal{P}_1$ and $\mathcal{P}_2$

The result of training the RC described in Eq 8.1 with the parameters given in Table 8.1 is shown in Fig. 8.4.

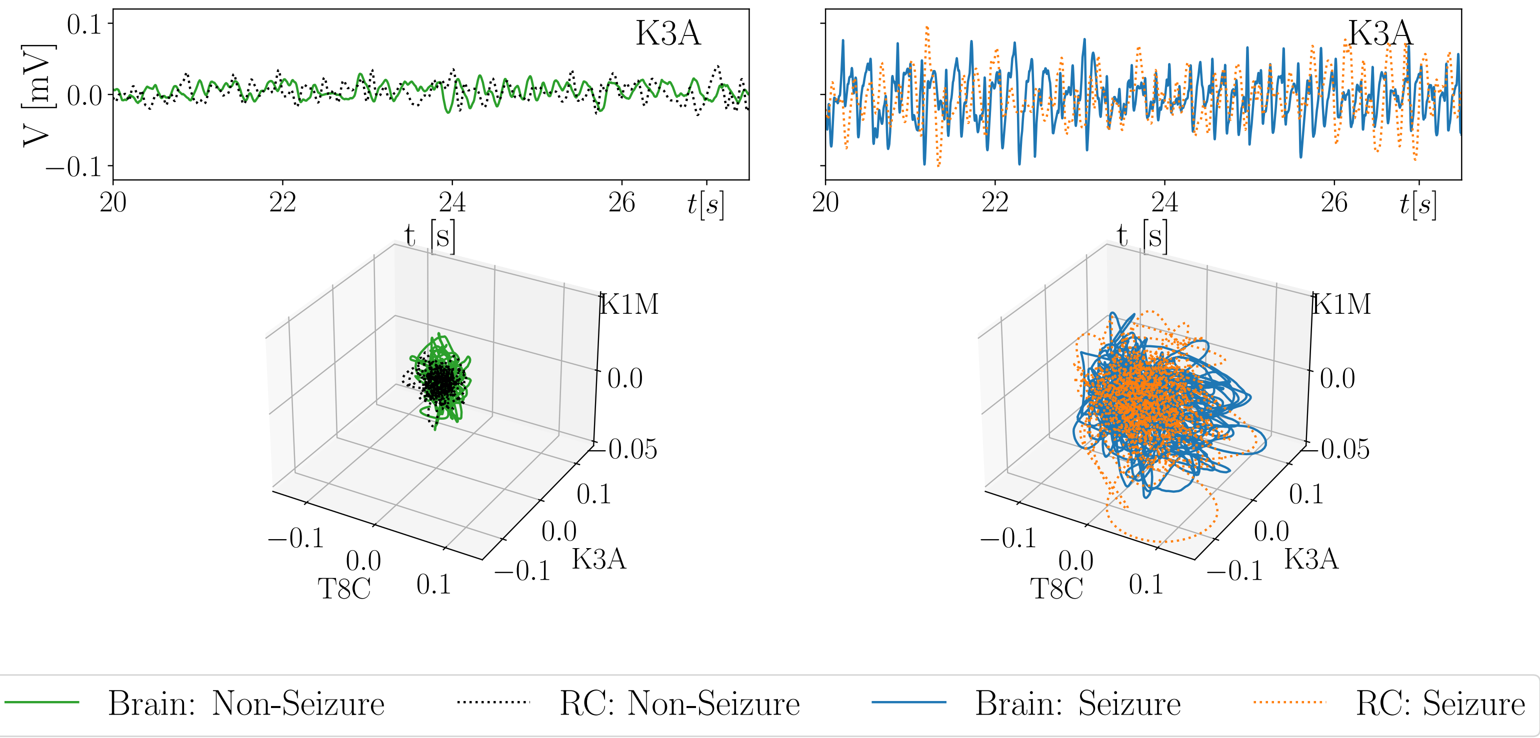


Figure 8.4: Multifunctional RC's reconstruction of the seizure and non-seizure attractors where t is the time after the prediction begins.

The dynamics of the trained multifunctional RC when it is initialised with either $\boldsymbol{r}_{\mathcal{S}_1}(t_{\text{train}})$ or $\boldsymbol{r}_{\mathcal{S}_2}(t_{\text{train}})$ is illustrated in Fig. 8.4. The bottom panels demonstrate that the multifunctional RC successfully reconstructs the dynamics of the non-seizure and seizure attractors for a long period of time.

The top panels in Fig. 8.4 illustrates a portion of the multifunctional RC's short term predictions of data from the K3A channel where the time is labelled here as the time after the RC begins its predictions from. These highlight that the RC closely imitates the dynamics of this variable during this period of time for both the non-seizure and seizure attractors. Although it is not shown here, this multifunctional RC was initialised from a number of random initial conditions and the non-seizure and seizure attractors are the only attractors which were found to exist in $\mathbb{P}$, in other words, there are no untrained attractors present after the training.

An extensive parameter sweep of the RC was conducted in order to find a combination of parameters which gave rise to multifunctionality. However, the large amount of overlap between the non-seizure and seizure neural data as seen in Fig. 8.3, presented similar difficulties to those experienced in Chapters 6 and 7 when the RC was tasked with solving the seeing double problem in the extreme case of $x_{cen} = 0$. The results presented in Figs. 7.1c and 7.4 provided a guideline for choosing the RC training parameters specified in Table 8.1.

8.3.3 Generating transitions between attractors

To generate transitions between the non-seizure and seizure attractors of the multifunctional RC, the same sequence of events which gave rise to the regime of chaotic itinerancy illustrated in Fig. 5.6 are followed here. In this case, in order to destabilise the multifunctional RC, the spectral radius, ρ, is increased from the value used in the training of 1.35 to 1.36. Note, it is only that $\mathbf{M}$ is rescaled here and the RC is not retrained at this new ρ value.

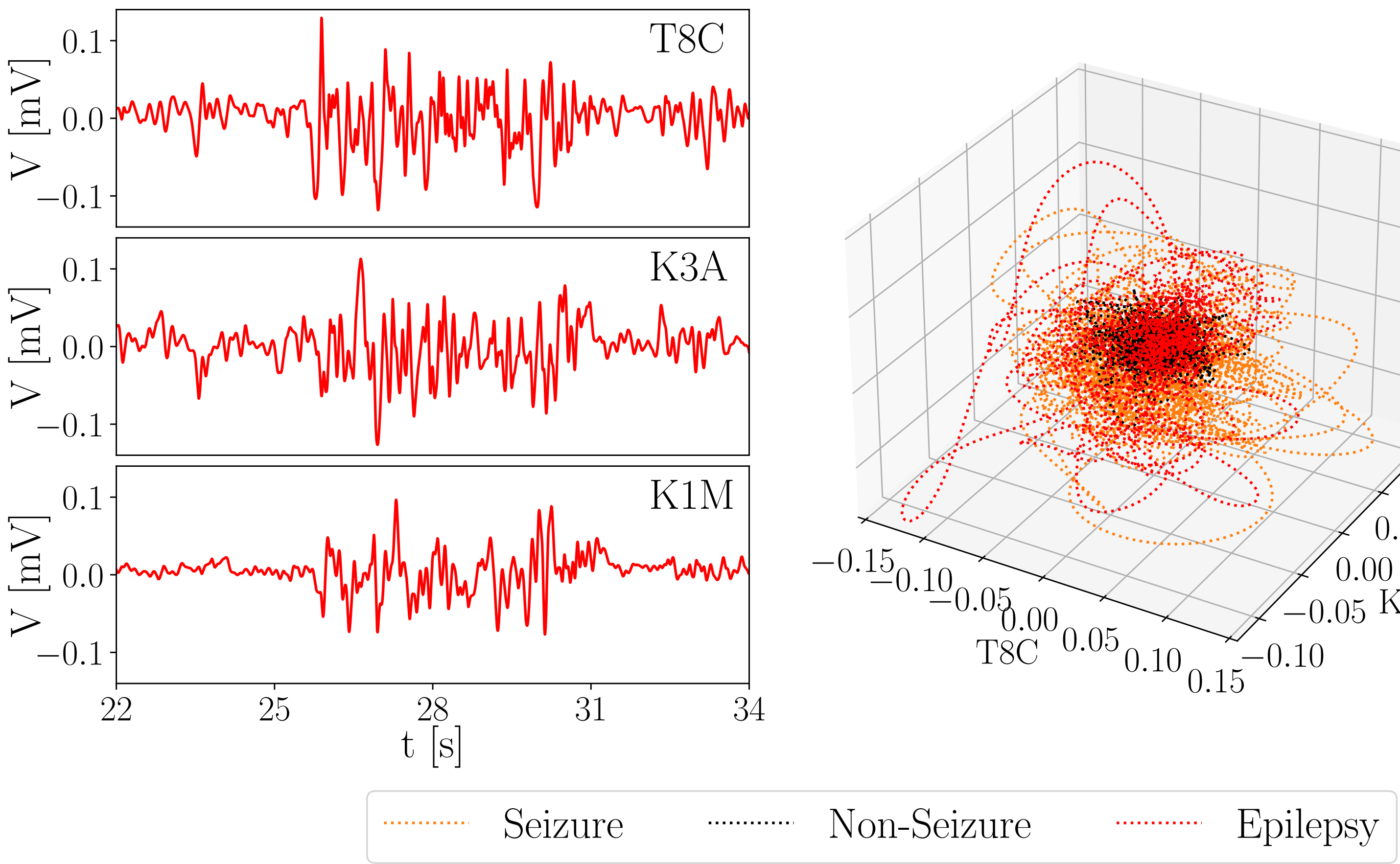


Figure 8.5: Illustrating a trajectory of the RC when exhibiting chaotic itinerancy in comparison to the multistable dynamics of the multifunctional RC where t is the time after the prediction begins.

The result of this is depicted in Fig. 8.5 which shows, in red, a trajectory of the RC that closely resembles the dynamical features of the training data. Here it can be seen that this small change in ρ is sufficient to destabilise the multifunctional RC and bring about chaotic itinerancy. The panels on the left hand side illustrate the RC's abrupt transition from the non-seizure to seizure and back to non-seizure attractor ruins. In the long-term, the length of time spent in seizure is characteristic of the experimental data, however, what is not shown here is that this property has a dependency on the choice of ρ.

The panel on the right hand side of Fig. 8.5 illustrates the itinerant trajectory overlaid on the original reconstructed attractors from the multifunctional RC. Here the RC's trajectory when visiting each attractor ruin closely resembles the dynamical features of the original coexisting attractors.

The best results were achieved by choosing the training parameters to be close to where chaotic itinerancy was first noted and the transitions began. While it was found that it is possible to achieve multifunctionality for slightly smaller values of ρ this required a larger increase in ρ in order for the RC to begin exhibiting chaotic itinerancy. This large gap between ρ values resulted in the respective attractors occupying larger and larger volumes of the state space before the transitions between them began. By training the RC with a ρ value just prior to the onset of chaotic itinerancy, this prevents the seizure and non-seizure attractor ruins from growing too large and retaining less features of the original training data.

8.4 Conclusions

The steps involved in training a RC that incorporates multifunctionality to successfully simulate the transitions between non-seizure and seizure dynamical regimes found in epileptic rodent brains are presented in this Chapter. To be more specific, a RC is trained using experimental neural voltage data obtained from GAERS to reconstruct a coexistence of non-seizure and seizure attractors. The dynamics of this multifunctional RC are destabilised in order for it to exhibit chaotic itinerancy. In this scenario the RC's dynamics captures the characteristics of the transitions observed in the experimental data as its state wanders chaotically between ruins of non-seizure and seizure attractors.

The results presented in this Chapter can also be extended to model the transitions between the non-seizure and seizure attractors using different mechanisms. For instance, by including an additional input variable in the trained multifunctional RC, a control signal can be designed to supply the RC with a sufficient input to force its state from one attractor to the other. Similarly, by driving the multifunctional RC with an external input, an excitable network attractor of the RC which describes the minimal input needed to switch from one attractor to the other could be established by extending the method presented in [123] to cases beyond fixed points. However this goes beyond the scope of the current Chapter.

Furthermore, as this RC successfully simulates characteristic features of the experimental neural voltage data from different rats, in future work the aim is to move into the 'patient specific' realm by modelling the dynamical traits of epilepsy in individual rats. Further ambitions of this work is to include additional 'non-seizure' attractors, such as the sleep state, in the training in order to determine if transitions can take place more easily between certain non-seizure attractors and seizure attractors.

As a final note, this Chapter marks a significant milestone in the story of multifunctionality as it establishes that not only is multifunctionality an interesting development to study from a theoretical perspective but it is also highly useful from an applications point of view. The results presented in this Chapter could not have been produced without the culmination of several key concepts and results developed throughout this Thesis.

Conclusion and Future Work

The results presented in this Thesis flow from a two-way street between neuroscientific observation and mathematical representation. Inspired by the ability of multifunctional neural networks to perform multiple tasks without changing any network connections, it is through recognising that these networks exploit a form of multistability that it becomes possible to translate multifunctionality from the biological to artificial neural networks via the reservoir computing ML paradigm.

The fundamentals of reservoir computing are outlined in Chapter 2 and the concept of 'generalised synchronisation' (GS) is used to describe the particular mechanisms that enable a reservoir computer (RC) of a particular form to learn how to reconstruct the long-term dynamics of a given attractor. In Chapter 3 the concept of GS is extended to cater for the case of training a RC to achieve multifunctionality by reconstructing a coexistence of several attractors. Here, the 'blending technique' is introduced as a means to train the RC for tasks involving multifunctionality. These concepts which underpin how a RC achieves multifunctionality are elucidated through several numerical experiments during Chapters 4 - 8.

More specifically, the initial focus of Chapter 4 is to test the flexibility of the blending technique by using it to train a RC to reconstruct a coexistence of chaotic attractors from a system that already exhibits multistability, two parameter settings of a system, and two different systems entirely. Upon the discovery of multiple other 'untrained attractors', attractors that exist in the state space of the trained RC but were not present during the training, the foucs shifts to exploring the relationship between these untrained attractors and the trained attractors using a bifurcation analysis.

Chapter 5 demonstrates the benefits of studying the effects of symmetries in a RC setup and serves as a route to better understand the fundamentals of this ML paradigm. Overall, the two-armed approach of analytical results together with numerical experiments provide a much greater insight towards the relationship between symmetry and multifunctionality in a RC. The inclusion of exotic cases like overlapping regions between attractors, chaotic itinerancy in the prediction state space, and achieving multifunctionality with four attractors further increases the robustness of the results presented in this Chapter and showcases some of the interesting dynamics a RC can be trained to simultaneously reconstruct.

The relationship between overlapping training data and multifunctionality comes to light in Chapter 6 by exploring the dynamics of a RC as it is trained to perform the 'seeing double' problem. The results presented in this Chapter reveal the particular bifurcations responsible for enabling a RC to achieve multifunctionality in a paradigmatic setting. Moreover, the conceptual framework that describes how RCs achieve multifunctionality outlined in Chapter 3 is based on the knowledge gained from the analytical and numerical results presented in this Chapter.

The seeing double problem is reinterpreted as a benchmark task for exploring the limits of multifunctionality across different RCs in Chapter 7. The results presented here provide some further guidance on how to train RCs to achieve multifunctionality in challenging circumstances and outline the key differences between how certain RCs achieve multifunctionality.

Multifunctionality opens up new application areas for RCs, for instance, in producing data-driven models of real-world phenomenon where multistability is thought to play a role. The first application of multifunctionality to a real-world problem is established in Chapter 8 where it is used to model seizure dynamics using experimental data from the brain of epileptic rodents.

While this Thesis introduces the theory of how RCs achieve multifunctionality, explores the dynamics of how multifunctionality is achieved in several numerical experiments, and lays the groundwork for future applications of multifunctionality, there are a growing number of question to answer and avenues for future work. Many of these questions and opportunities for future work are highlighted throughout this Thesis, the following discussion provides a general overview on how to address some of these.

One of the most outstanding questions that remains is, how much multifunctionality can a given RC achieve? From the results presented in this Thesis, it is clear that there's at least two factors to consider here, the number of attractors and the relationship between these attractors. The current record for the most amount of attractors reconstructed by a multifunctional RC is held by Joschka Herteux who trained a RC to reconstruct a coexistence of seven chaotic attractors as shown in [175]. Acknowledging that it is possible to compose systems with 'infinitely many' coexisting chaotic attractors like in [176], then the same could be explored for a RC. The benefits of pursuing such endeavours in terms of memory recall and computational efficiency are discussed in Sec. 3.4. At the same time, Fig. 7.1c illustrates that even when the relationship between two limit cycles is highly complicated then RCs with much higher dimensions are required in order for multifunctionality to be achieved.

A key element of this Thesis is the study of the interaction between untrained attractors and trained attractors with examples of these presented in Chapters 2, 4, and 6. Based on these studies it seems that untrained attractors prepare the state space of the RC for the arrival of the trained attractors. However, further investigations are required in order to establish the root and the role of untrained attractors in a RC.

While the bifurcation analysis tools used in this Thesis provide significant insight into how RCs learn and perform certain tasks and their interaction with untrained attractors, it is likely that only half the story is being told as the unstable solutions are not accounted for due to the inherent limitations of the approach used to explore these dynamics. As mentioned in Chapter 4, these issues can be alleviated by adapting well-known continuation software like AUTO [134] to the case of reservoir computing. Bifurcation analysis on ANNs similar to the CT-RC studied throughout the majority of this Thesis have so far mainly concentrated low dimensional ANNs such as the work of Doya in [177], Beer in [178,179], and Ceni *et al.* in [123] where the results from a bifurcation analysis of the LI-RC provides the insight needed to solve the flip-flop problem. By conducting a bifurcation analysis of low-dimensional RCs and studying how the dynamics of the RC changes as the dimension increases may provide further insight into the behaviour of higher dimensional RCs and establish how much multifunctionality a given RC can have.

From an applications perspective, multifunctional reservoir computing can also be used to model further neurological conditions involving the transition between different regimes in competition with one another. Furthermore, using a multifunctional RC to bring about transitions between states in an autonomous system is not limited to modelling neural dynamics. In principle, when there is an interaction between a number of distinct modes of operation, the modelling method outlined in this paper is applicable.

It would also be interesting to investigate if multifunctionality can also be translated to Physical RCs like those mentioned in Sec. 2.3.5 to increase the range of their applications. The work presented in this Thesis may be of particular interest to researchers designing on-chip RCs to be deployed in embedded devices as some applications of these may benefit from incorporating multifunctionality.

All in all, the work presented in this Thesis indicates that if other hypotheses or known facets of the brain can be articulated using concepts from dynamical systems, then therein lies the potential to portray it artificially. Moreover, the brain may also hold many more secrets that can push the boundaries of what a given ANN can be trained to achieve.

Appendix A

The Explainability Elephant

Confronting the divergence between the ability and explainability of AI

Throughout history, humans have developed a multitude of tools and machines which have changed our relationship with the world. From the earliest implements used by our ancient ancestors to the technologies of today, mathematics has played a crucial role in the development and understanding of these inventions. However, the rate at which highly disruptive machine learning innovations have advanced in recent times has proven to be difficult for mathematics to keep pace with explaining how these machines learn. The growing need to address this widening gap and a means to tackle this problem become clear through the topics considered in the following discussion that's directed to a more general audience.

A.1 From picking stones to the Singularity

While the first known tools used by our early ancestors mainly consisted of carved stones shaped for specific tasks [180], these innovations arose from our growing awareness of how we can interact with our immediate surroundings. However, in order to perform tasks beyond the physiological and computational capabilities of the body and mind, this ultimately required a basic understanding of several physical phenomena and mathematical concepts.

This story is still more or less the same over 3 million years later. For instance, the invention of the steam engine during the Industrial Revolution required a deep understanding of thermodynamics and mechanics. These years of tremendous scientific and technological advances marked a turning point in human history as machines began to replace us in jobs previously done by hand, ergo, resulting in extensive social and economic changes.

In more modern times, the increasing power and widespread use of computers has had an immensely profound impact on human society. From the way we communicate and work, to how we enlighten and entertain ourselves, these information processing machines have changed nearly every aspect of life. However, it is only in last number of years that questions on whether computers can think or could acquire a form of intelligence or even gain consciousness have become commonplace and entered the collective societal conscience thanks to advances in the research area of 'Artificial Intelligence' (AI).

Today, AI is recognised as an umbrella term referring to a suite of technologies that mimic and outperform humans on a growing list of tasks.

In recent years, there has been an explosion of interest in these technologies. This is largely due to the enormous progress made across several branches of AI, in particular, in the area of Machine Learning (ML) with the development of new techniques for training 'artificial neural networks'[1] (ANNs) to solve sophisticated problems. From aiding professionals in healthcare [9, 10] and agriculture [11–13], to advances in autonomous vehicles [14–16], and beating world champions in the game Go [17], various designs of ANNs have quickly become a dominating factor in driving forward technological innovation.

These ML breakthroughs have generated immense economic growth and companies around the world are looking to incorporate AI into multiple different products and services. AI based technologies are projected to expand the world economy by up to 14% during the present decade, however, this progress may sweep across many professions like a double-edged sword as significant job displacement and creation are also expected [181].

Despite any of these potential opportunities, with every transformational discovery this brings with it many new challenges which cannot be wholly anticipated. To put this in greater perspective, let us turn to Seneca, the Stoic philosopher, who said, "*non est ad astra mollis e terris via*" ("there is no easy way from the Earth to the stars"). In the case of AI and ML, there

[1] Mathematical abstractions of the basic building blocks of biological neural networks.

is still much work to be done, not only in terms of assessing the long-term reliability of artificially intelligent systems but also in developing certification guidelines, before a widespread adoption in industry can occur [182].

Central to these concerns is the lack of explainability on how many of these ANNs operate [43–45]. In response to this, there is a growing emphasis placed on the advancement of 'Explainable AI' in order to construct a Rosetta stone that enable humans to understand the decisions or predictions made by an artificially intelligent system. However, what is just as troubling is that, despite the remarkable achievements made in AI, there is a tendency to 'shoot first and ask questions later' as many ANNs are considered to be 'black-boxes'[2]. This oftentimes results in little being known about the internal mechanisms responsible for enabling learning, preventing learning, or producing unintended biases or functionalities.

As the gap between advances in AI and our awareness of how these machines operate continues to widen, and the line between human and machine intelligence continues to blur, where does this momentum lead us to? It is natural to imagine a stage whereby this line between humans and machines becomes sharply redefined. This event, is what John von Neumann conceived of as the '*Singularity*' - a point in time at which the intelligence of machines surpasses human intelligence and fundamentally changes the nature of civilisation. Discussions on this event has entered into the public domain thanks to books such as Ray Kurzweil's, *The Singularity is Near* [183], where he outlines his reasons for predicting the Singularity to occur by 2045.

A.2 A Singularity survival guide

For many years the woes of wild post-Singularity visions of human civilisation have captivated audiences worldwide. Several Sci-Fi fiascoes have asked questions such as, whether humans will begin crusades against computers and outlaw AI as in Denis Villeneuve's recent adaption of Frank Herbert's *Dune*,

[2] A black-box is a system whereby there is nothing known about how the system produces a particular output for a given input.

reverse roles and become slaves to machines like in the Wachowski's *Matrix*, fall victim to rogue robots and AIs such as HAL 9000 in Stanley Kubrick's *2001: A Space Odyssey*, James Cameron's *Terminator*, Ava in Alex Garland's *Ex Machina*, or attempt to coexist amongst our sentient creations like Ridley Scott's dystopian dream of Los Angeles in *Blade Runner* that ultimately juxtaposes what it means to be human in the first place. These fantastical figments manifest from subtle perturbations of technological potentialities from our world and failed attempts to create utopia resulting in either radical societal shifts or collapse [3]. All in all, these nightmares serve as warning signals towards the importance of building 'our final invention'[4] correctly.

In reality, it is difficult to fully imagine what our place in the world will entirely look like after human-level AI is achieved and cognition as we know it is liberated from its current biological limitations. However, many psychoanalysts are questioning not only what will be the societal impact of an increasing dependence on AI but also how our present belief systems will function if the mind and machines coalesce in the 'wired brain'[5]. For instance, in Slavoj Žižek's recent book, *Hegel in a Wired Brain* [186], he outlines how our current understanding of human ideology as well as different schools of philosophy require radical rethinking in a post-Singularity society consisting of multiple competing forms of intelligent agents such as humans, post-humans[6], human developed AI, and non-human developed AI.

At the same time, in Mark O'Connell's book, *To Be a Machine* [187], he recounts of several encounters with advocates of the *Transhumanist*[7] move-

---

[3]Through William Gibson's 1984 debut novel, *Necromancer*, he's credited with identifying that these artistic movements have spawned a new subgenre in Sci-Fi known as *Cyberpunk*, where questions on humanity's place in a technologically advanced transcultural ecosystem are explored [184].

[4]A term coined by James Barrat to describe the potential future impact of AI in his book, *Our Final Invention: Artificial Intelligence and the End of the Human Era* [185].

[5]A term synonymous with a brain-computer interface (BCI), a device implanted in the brain which creates a dialogue between signals in the brain and, for example, a computer, robotic limb, or another BCI.

[6]A person or an entity that exists in a state beyond being human resulting from potential enhancements to human nature generated through science and technological developments.

[7]A movement, popularised by biologist Julian Huxley (brother of the author Aldous

ment who proclaim that our destiny is to transcend to post-human lifeforms, extract ourselves from the biological bounds imposed on us by nature, and to ultimately return to our place in the Garden of Eden. O'Connell details how certain body hacking fanatics claim to have become cyborgs[8] by implanting antennae and sensors into themselves with many more enthusiasts counting down the days to the Singularity fueled by hopes of future technologies which can be used to avoid death by uploading their mind to a computer.

Regardless of what side anyone is on, the bottom line is that there is an increasing amount of ML applications creeping into components of everyday life and there is still relatively little known about how these machines actually learn and perform certain tasks. Therefore, as our dependence on AI based technologies continues to grow, it is paramount that we practice some prescience and develop the mathematics required to describe the dynamics of these machines before their behaviour becomes incomprehensible to us. Žižek can offer further guidance on the mindset needed to confront these issues with his final message in *First as Tragedy, Then as Farce* [188], by realising that "we are the ones we have been waiting for". Through this it becomes clear that, even though the promise of AI is to improve the human condition it may well be a Trojan Horse waiting at the gates, and the only way to prevent Sci-Fi apocalypse dreams from becoming a reality is to perceive it as our fate and to act proactively in order to avoid it.

A.3 Follow the dynamical road

Fortunately, it is possible to simultaneously push the boundaries of AI and address the explainability elephant in the room by employing the interdisciplinary mathematical language of dynamical systems.

In a nutshell, dynamical systems is a branch of mathematics that focuses

Huxley and step-brother of the Nobel Prize-winning physiologist Andrew Huxley), which is predicated on the idea that we are already machines that just happen to run on what Earth provides and that we can and should use technology to change the basic aspects of the human condition, such as the certainty of death, to ultimately become immortal.

[8] A cybernetic organism with organic and biomechatronic body parts.

on studying the behaviour/dynamics of systems that evolve over time according to a set of equations. From Henri Poincaré's pioneering approach to the three-body problem [189], to Edward Lorenz's discovery of 'deterministic nonperiodic flow' [75] and the boom years of nonlinear dynamics and chaos that followed [190], to the broad-spectrum of phenomena studied by *dynamicists*[9] worldwide today, our growing knowledge of dynamical systems has enabled us to navigate our way to new frontiers of science.

Therefore, by describing an artificially intelligent system as a dynamical system then the same set of mathematical tools, that have dramatically improved our understanding of the world around us, can in principal be used to explain how these machines learn.

In this Thesis, several concepts from dynamical systems are used to explore how a 'reservoir computer' (RC) [28–30] learns and solves particular tasks. Reservoir Computing is just one ML approach which can be rigorously assessed through the lens of dynamical systems as a RC can be realised as a type of ANN that is also a dynamical system. The success of examining RCs from a dynamical systems perspective is well documented in the literature and there has been many key advances made which help to 'open the black-box' such as [38–42]. The results presented in this Thesis help to crack open the black-box even further by identifying, in certain circumstances, the particular bifurcations[10] a RC goes through that are responsible for enabling learning, preventing learning, and producing unintended functionalities.

Another benefit of working within this dynamical systems viewpoint of ML is that well-known concepts in dynamical systems can also be exploited to establish new ways in which a machine can learn [114, 115]. As this Thesis demonstrates, this also facilitates the integration of known dynamical features of the brain into ML environments which reinforces the bridge between the biological and artificial neural networks in order to develop artificially intelligent systems that are more closely aligned with operational characteristics of the brain.

[9]A person who investigates and researches dynamics.

[10]A qualitative change in the behaviour of a dynamical system that occurs by varying a parameter of the system.

Appendix B

RC calculations

B.1 Computing the Jacobian matrix

The Jacobian matrix is calculated by evaluating the derivative of the predicting RC from Eq. (2.19) with respect to the state of the RC, $\hat{\boldsymbol{r}}(t) \in \mathbb{R}^N$. In other words,

$$\mathbf{J}(t) = \frac{\partial}{\partial \hat{\boldsymbol{r}}} \dot{\hat{\boldsymbol{r}}}(t), \tag{B.1}$$

where $\boldsymbol{J}(t)$ is the Jacobian matrix of the predicting RC evaluated at time t. Applying this to Eq. (2.19) results in,

$$\begin{aligned} &\frac{\partial}{\partial \hat{\boldsymbol{r}}} \left\{ \gamma \left[-\hat{\boldsymbol{r}}(t) + \tanh\left(\mathbf{M}\, \hat{\boldsymbol{r}}(t) + \sigma \mathbf{W}_{in} \mathbf{W}_{out}\, \boldsymbol{q}(\hat{\boldsymbol{r}}(t)) \right) \right] \right\} = \\ &-\gamma\, \mathbf{I}_N + \gamma \frac{\partial}{\partial \hat{\boldsymbol{r}}} \tanh\left(\mathbf{M}\, \hat{\boldsymbol{r}}(t) + \sigma \mathbf{W}_{in} \mathbf{W}_{out}\, \boldsymbol{q}(\hat{\boldsymbol{r}}(t)) \right). \end{aligned} \tag{B.2}$$

where $\mathbf{M} \in \mathbb{R}^{N \times N}$, $\mathbf{W}_{in} \in \mathbb{R}^{N \times D}$, $\mathbf{W}_{out} \in \mathbb{R}^{D \times 2N}$, $\boldsymbol{q}(\hat{\boldsymbol{r}}(t))$ is defined as in Eq. (2.13), and $\mathbf{I}_N$ denotes an $N \times N$ identity matrix.

In order to make further simplifications of Eq. (B.2), it is necessary to use the chain-rule, $\frac{\partial}{\partial \boldsymbol{x}} f(g(\boldsymbol{x})) = \frac{\partial}{\partial g(\boldsymbol{x})} f(g(\boldsymbol{x})) \times \frac{\partial}{\partial \boldsymbol{x}} g(\boldsymbol{x})$. In the case of the second term in Eq. (B.2), by using the chain-rule it follows that,

$$\begin{aligned} &\frac{\partial}{\partial \hat{\boldsymbol{r}}} \tanh\left(\mathbf{M}\, \hat{\boldsymbol{r}}(t) + \sigma\, \mathbf{W}_{in} \mathbf{W}_{out}\, \boldsymbol{q}(\hat{\boldsymbol{r}}(t)) \right) = \\ &\frac{\partial}{\partial \left(\mathbf{M}\, \hat{\boldsymbol{r}}(t) + \sigma \mathbf{W}_{in} \mathbf{W}_{out}\, \boldsymbol{q}(\hat{\boldsymbol{r}}(t)) \right)} \tanh\left(\mathbf{M}\, \hat{\boldsymbol{r}}(t) + \sigma \mathbf{W}_{in} \mathbf{W}_{out}\, \boldsymbol{q}(\hat{\boldsymbol{r}}(t)) \right) \\ &\times \frac{\partial}{\partial \hat{\boldsymbol{r}}} \left(\mathbf{M}\, \hat{\boldsymbol{r}}(t) + \sigma \mathbf{W}_{in} \mathbf{W}_{out}\, \boldsymbol{q}(\hat{\boldsymbol{r}}(t)) \right). \end{aligned} \tag{B.3}$$

Focusing first on the derivative of the tanh $(\cdot)$ term in Eq. (B.3), the following substitution, $\boldsymbol{s}(t) = \mathbf{M}\,\hat{\boldsymbol{r}}(t) + \sigma\mathbf{W}_{in}\mathbf{W}_{out}\,\boldsymbol{q}(\hat{\boldsymbol{r}}(t)) = (s_1\,\ldots\,s_N)$, is made for a greater ease of computation. This derivative term is now written as,

$$\frac{\partial}{\partial \boldsymbol{s}}\tanh\boldsymbol{s} = \begin{bmatrix} \frac{\partial}{\partial s_1}\tanh s_1 & \cdots & \frac{\partial}{\partial s_N}\tanh s_1 \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial s_1}\tanh s_N & \cdots & \frac{\partial}{\partial s_N}\tanh s_N \end{bmatrix}. \tag{B.4}$$

All off diagonal elements of the above matrix are 0 which allows Eq. (B.4) to be written as,

$$\frac{\partial}{\partial \boldsymbol{s}}\tanh\boldsymbol{s} = \mathrm{diag}\left(\frac{\partial}{\partial s_1}\tanh s_1\ldots\frac{\partial}{\partial s_N}\tanh s_N\right), \tag{B.5}$$

where diag $(\cdot)$ denotes a diagonal matrix constructed from the elements of a given vector. By computing the derivative of the above, it follows that,

$$\begin{aligned} \mathrm{diag}\left(\frac{\partial}{\partial \boldsymbol{s}}\tanh\boldsymbol{s}\right) &= \mathrm{diag}\left(\frac{1}{\cosh\boldsymbol{s}^2}\right), \\ &= \mathrm{diag}\left(\frac{1}{\cosh\left(\ \mathbf{M}\hat{\boldsymbol{r}}(t) + \sigma\mathbf{W}_{in}\mathbf{W}_{out}\boldsymbol{q}(\hat{\boldsymbol{r}}(t))\ \right)^2}\right). \end{aligned} \tag{B.6}$$

The other derivative term in Eq. (B.3), $\frac{\partial}{\partial\hat{\boldsymbol{r}}}\left(\mathbf{M}\,\hat{\boldsymbol{r}}(t) + \sigma\mathbf{W}_{in}\mathbf{W}_{out}\,\boldsymbol{q}(\hat{\boldsymbol{r}}(t))\right)$, is evaluated as,

$$\begin{aligned} &\frac{\partial}{\partial\hat{\boldsymbol{r}}}\mathbf{M}\,\hat{\boldsymbol{r}}(t) + \frac{\partial}{\partial\hat{\boldsymbol{r}}}\sigma\mathbf{W}_{in}\mathbf{W}_{out}\,\boldsymbol{q}(\hat{\boldsymbol{r}}(t)), \\ &= \mathbf{M}\,\frac{\partial}{\partial\hat{\boldsymbol{r}}}\hat{\boldsymbol{r}}(t) + \sigma\mathbf{W}_{in}\mathbf{W}_{out}\,\frac{\partial}{\partial\hat{\boldsymbol{r}}}\boldsymbol{q}(\hat{\boldsymbol{r}}(t)), \\ &= \mathbf{M}\,\mathbf{I}_N + \sigma\mathbf{W}_{in}\mathbf{W}_{out}\begin{bmatrix} \mathbf{I}_N \\ 2\left(\mathrm{diag}\left(\hat{\boldsymbol{r}}(t)\right)\right) \end{bmatrix}. \end{aligned} \tag{B.7}$$

Combining the simplifications made in Eqs. (B.6) and (B.7) and substituting these into Eq. (B.2) allows the Jacobian of the predicting RC at a given time t to be written as,

$$\begin{aligned} \mathbf{J}(t) = -\gamma\mathbf{I}_N + \gamma\,\mathrm{diag}\left(\frac{1}{\cosh\left(\mathbf{M}\hat{\boldsymbol{r}}(t) + \sigma\mathbf{W}_{in}\mathbf{W}_{out}\boldsymbol{q}(\hat{\boldsymbol{r}}(t))\right)^2}\right)\mathbf{M} \\ + \gamma\sigma\,\mathrm{diag}\left(\frac{1}{\cosh\left(\mathbf{M}\hat{\boldsymbol{r}}(t) + \sigma\mathbf{W}_{in}\mathbf{W}_{out}\boldsymbol{q}(\hat{\boldsymbol{r}}(t))\right)^2}\right)\mathbf{W}_{in}\mathbf{W}_{out}\begin{bmatrix} \mathbf{I}_N \\ 2\left(\mathrm{diag}\left(\hat{\boldsymbol{r}}(t)\right)\right) \end{bmatrix}. \end{aligned}$$

Appendix C

Numerical methods

C.1 Lyapunov exponents

In 1892, in his doctoral thesis, *The general problem of the stability of motion* [191], Aleksandr Lyapunov introduced several groundbreaking ideas to investigate stability in differential equations. These concepts, including Lyapunov functions and Lyapunov exponents, are collectively known as Lyapunov Stability Theory.

Since their discovery, Lyapunov exponents (LE) have proven to be one of the most useful tools in the analysis of dynamical systems, in particular for quantitatively describing the behaviour of a given system. These exponents are an exponential measure of divergence or convergence of nearby orbits in the systems state space. If the largest Lyapunov exponent (LLE), λ_{max}, of a given attractor is:

- Positive, implies the system exhibits chaos.
- Zero, suggests the system is at a bifurcation point or is on a stable limit cycle.
- Negative, indicates that the system is on a stable fixed point equilibrium.

The following process is used to compute the largest Lyapunov exponent of a given continuous-time dynamical system,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t)\right). \tag{C.1}$$

After the system has settled down to some attractor, $\mathcal{A}$, after some time t when initialised with, $\boldsymbol{x}(0) = \boldsymbol{x}_{\mathcal{A}}$, then the solution of Eq. (C.1) can be written as $\boldsymbol{x}_{\mathcal{A}}(t)$.

To test the local stability of $\boldsymbol{x}_{\mathcal{A}}(t)$ and compute the LE's to determine if nearby trajectories converge to or diverge from $\boldsymbol{x}_{\mathcal{A}}(t)$, the system is linearised about $\boldsymbol{x}_{\mathcal{A}}(t)$ and the evolution of a small perturbation vector $\boldsymbol{p}(t)$ beginning nearby $\boldsymbol{x}_{\mathcal{A}}(t)$ is written as,

$$\boldsymbol{x}(t) = \boldsymbol{x}_{\mathcal{A}}(t) + \boldsymbol{p}(t). \tag{C.2}$$

Substituting the above into Eq. (C.1) results in,

$$\frac{d}{dt}\boldsymbol{x}_{\mathcal{A}}(t) + \frac{d}{dt}\boldsymbol{p}(t) = \boldsymbol{f}\left(\boldsymbol{x}_{\mathcal{A}}(t) + \boldsymbol{p}(t)\right), \tag{C.3}$$

$$= \boldsymbol{f}\left(\boldsymbol{x}_{\mathcal{A}}(t)\right)\mathbf{J}\left(\boldsymbol{x}_{\mathcal{A}}(t)\right)\boldsymbol{p}(t) + \mathcal{O}\left(\boldsymbol{p}^2(t)\right), \tag{C.4}$$

where $\mathbf{J}\left(\boldsymbol{x}_{\mathcal{A}}(t)\right)$ is the Jacobian matrix of the system described in Eq. (C.1) evaluated at $\boldsymbol{x}_{\mathcal{A}}(t)$. The evolution of the perturbation vector $\boldsymbol{p}(t)$ from initial condition, $\boldsymbol{p}(0) = \boldsymbol{p}_0$, can be written as,

$$\dot{\boldsymbol{p}}(t) = \mathbf{J}\left(\boldsymbol{x}_{\mathcal{A}}(t)\right)\boldsymbol{p}(t). \tag{C.5}$$

To compute the LLE of the system, solutions of both Eqs. (C.1) and (C.5) are calculated simultaneously, first allowing Eq. (C.1) to settle to an attractor and then letting $\boldsymbol{p}(t)$ grow for some short interval of time, t' to avoid $\boldsymbol{p}(t)$ growing too large. The following is then computed,

$$l^1 = \frac{1}{t'}\ln \boldsymbol{p}(t'), \tag{C.6}$$

and $\boldsymbol{p}(t')$ is rescaled as $\boldsymbol{p}(t') \to \dfrac{\boldsymbol{p}(t')}{|\boldsymbol{p}(t')|}$ preserving the direction of the vectors orientation of convergence/divergence from the attractor. This method is repeated multiple times to calculate $l^2, l^3, \ldots, l^n$ with n chosen sufficiently large for to converge to the LLE, λ_{max}, where,

$$\lambda_{max} \approx \frac{1}{n}\sum_{i=1}^{n} l^i. \tag{C.7}$$

C.2 Floquet multipliers

In 1883, Gaston Floquet introduced the concept of Floquet multipliers to analyse the periodic behaviour exhibited certain differential equations [192]. Floquet multipliers characterise the stability of a dynamical system's response to a periodic driving input signal. In addition to this, and in contrast to Lyapunov exponents, through tracking the evolution of the Floquet multipliers in response to changes in the dynamical system's parameters, it is also possible to determine what type of bifurcation a dynamical system undergoes at certain parameter values.

The Floquet multipliers are the eigenvalues of the 'monodromy matrix', $\mathbf{Q}$, which is the solution of,

$$\dot{\mathbf{Q}}(t) = \mathbf{J}(t)\mathbf{Q}(t), \quad \mathbf{Q}(0) = \boldsymbol{I}_N, \tag{C.8}$$

after one period, T. $\mathbf{J}(t)$ is the Jacobian matrix of a dynamical system which exhibits periodic dynamics. $\boldsymbol{I}_N$ is the identity matrix of size $N \times N$.

The eigenvalues of $\mathbf{Q}$ provides information on the stability of the driven system. Given a periodic attractor, if the absolute value of any Floquet multiplier, μ_i, is > 1 then the attractor is unstable. On the other hand, if the absolute value of the largest Floquet multiplier, μ_1, is 1 and all other μ_i's have absolute value < 1 then the attractor stable. There will always be one eigenvalue of $\mathbf{Q}$ which equals 1, this is called the 'trivial' Floquet multiplier.

Further information about the dynamics of the system can also be determined from examining the evolution of the Floquet multipliers in correspondence to changes in system parameters. For instance, as a given system parameter is varied, if the largest Floquet multiplier, written as $\mu_1 = x + i\,y$, changes such that,

$$\begin{array}{ccc} x > 1,\ y = 0 & \Longrightarrow & \text{saddle-node/symmetry-breaking bifurcation,} \\ x < -1,\ y = 0 & \Longrightarrow & \text{period-doubling bifurcation,} \\ |x + i\,y| > 1,\ x, y \neq 0 & \Longrightarrow & \text{Hopf/Torus bifurcation.} \end{array}$$

The level of insight which Floquet multipliers provide about dynamical systems in the form of RCs is highly valuable. Not only do the Floquet multipliers identify when a RC's response to a periodic driving signal is stable, and therefore when attractor reconstruction is possible, but also the particular bifurcation through which attractor reconstruction is achieved.

In order to compute $\mathbf{Q}$ in the case of the CT-RC presented in Eq. 2.10, its dynamics over one period, T, is recorded. In this instance, $\mathbf{J}(t)$ is the Jacobian matrix of the predicting RC defined in Eq. (2.19) which is computed in Appendix B.1.

References

[1] Richard Taylor. *Scientific memoirs, selected from the transactions of foreign academies of science and learned societies, and from foreign journals*, volume 3. R. and JE Taylor, 1843.

[2] George Boole. *An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities*, volume 2. Walton and Maberly, 1854.

[3] Alan Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 41(2):230–267, 1936.

[4] Andrew Hodges. *Alan Turing: The Enigma*. Princeton University Press, 2014.

[5] Apollodorus and Robin Hard. *The Library of Greek Mythology*. Oxford World's Classics - Oxford University Press. Oxford University Press, 1998.

[6] Alan Turing. Computing machinery and intelligence. *Mind*, 59(236):433, 1950.

[7] John McCarthy, Marvin L Minsky, Nathaniel Rochester, and Claude E Shannon. A proposal for the Dartmouth Summer Research Project on Artificial Intelligence. *AI magazine*, 27(4):12–12, 2006.

[8] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[9] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

[10] Pranav Rajpurkar, J. Irvin, Kaylie Zhu, B. Yang, Hershel Mehta, T. Duan, D. Ding, Aarti Bagul, C. Langlotz, K. Shpanskaya, M. Lungren, and A. Ng. Chexnet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. *arXiv preprint arXiv:1711.05225*, 2017.

[11] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419, 2016.

[12] Konstantinos P Ferentinos. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145:311–318, 2018.

[13] Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational intelligence and neuroscience*, 2016, 2016.

[14] Lex Fridman, Daniel E Brown, Michael Glazer, William Angell, Spencer Dodd, Benedikt Jenik, Jack Terwilliger, Aleksandr Patsekin, Julia Kindelsberger, Li Ding, et al. MIT advanced vehicle technology study: Large-scale naturalistic driving study of driver behavior and interaction with automation. *IEEE Access*, 7:102021–102038, 2019.

[15] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Mon-

fort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[16] Brody Huval, Tao Wang, S. Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, M. Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, Fernando A. Mujica, A. Coates, and A. Ng. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015.

[17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[20] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[22] Kevin L Briggman and WB Kristan Jr. Multifunctional pattern-generating circuits. *Annual review of neuroscience*, 31:271–294, 2008.

[23] George J Mpitsos and Christopher S Cohan. Convergence in a distributed nervous system: Parallel processing and self-organization. *Journal of neurobiology*, 17(5):517–545, 1986.

[24] Peter A Getting. Emerging principles governing the operation of neural networks. *Annual review of neuroscience*, 12(1):185–204, 1989.

[25] Patsy S Dickinson. Interactions among neural networks for behavior. *Current opinion in neurobiology*, 5(6):792–798, 1995.

[26] Eve Marder and Ronald L Calabrese. Principles of rhythmic motor pattern generation. *Physiological reviews*, 76(3):687–717, 1996.

[27] SP Lieske, M Thoby-Brisson, P Telgkamp, and JM Ramirez. Reconfiguration of the neural network controlling multiple breathing patterns: eupnea, sighs and gasps. *Nature neuroscience*, 3(6):600, 2000.

[28] David Verstraeten, Benjamin Schrauwen, Michiel d'Haene, and Dirk Stroobandt. An experimental unification of reservoir computing methods. *Neural Networks*, 20(3):391–403, 2007.

[29] Herbert Jaeger. The 'echo state' approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148, 01 2001.

[30] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

[31] Andrew Flynn, Vassilios A Tsachouridis, and Andreas Amann. Multifunctionality in a reservoir computer. *Chaos*, 31(1):013125, 2021.

[32] Andrew Flynn, Joschka Herteux, Vassilios A Tsachouridis, Christoph Räth, and Andreas Amann. Symmetry kills the square in a multifunctional reservoir computer. *Chaos*, 31(7):073122, 2021.

[33] Andrew Flynn, Oliver Heilmann, Daniel Köglmayr, Vassilios A. Tsachouridis, Christoph Räth, and Andreas Amann. Exploring the limits of multifunctionality across different reservoir computers. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022.

[34] Andrew Flynn, Vassilios A Tsachouridis, and Andreas Amann. Seeing double with a multifunctional reservoir computer. *arXiv preprint arXiv:2305.05799*, 2023.

[35] Ashesh Chattopadhyay, Pedram Hassanzadeh, and Devika Subramanian. Data-driven predictions of a multiscale lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Processes in Geophysics*, 27(3):373–389, 2020.

[36] Pantelis R Vlachas, Jaideep Pathak, Brian R Hunt, Themistoklis P Sapsis, Michelle Girvan, Edward Ott, and Petros Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191–217, 2020.

[37] Alexandre Variengien and Xavier Hinaut. A journey in ESN and LSTM visualisations on a language task. *arXiv preprint arXiv:2012.01748*, 2020.

[38] David Sussillo and Omri Barak. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, 25(3):626–649, 2013.

[39] Lyudmila Grigoryeva and Juan-Pablo Ortega. Echo state networks are universal. *Neural Networks*, 108:495–508, 2018.

[40] Thomas L Carroll. Do reservoir computers work best at the edge of chaos? *Chaos*, 30(12):121109, 2020.

[41] Allen Hart, James Hook, and Jonathan Dawes. Embedding and approximation theorems for echo state networks. *Neural Networks*, 128:234–247, 2020.

[42] Pietro Verzelli, Cesare Alippi, and Lorenzo Livi. Learn to synchronize, synchronize to learn. *Chaos*, 31(8):083119, 2021.

[43] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[44] Ribana Roscher, Bastian Bohn, Marco F Duarte, and Jochen Garcke. Explainable machine learning for scientific insights and discoveries. *IEEE Access*, 8:42200–42216, 2020.

[45] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion*, 58:82–115, 2020.

[46] Benjamin Schrauwen, Marion Wardermann, David Verstraeten, Jochen J Steil, and Dirk Stroobandt. Improving reservoirs using intrinsic plasticity. *Neurocomputing*, 71(7-9):1159–1171, 2008.

[47] Guillermo B Morales, Claudio R Mirasso, and Miguel C Soriano. Unveiling the role of plasticity rules in reservoir computing. *Neurocomputing*, 461:705–715, 2021.

[48] Jacob Morra and Mark Daley. A Fully-Connected neural network derived from an electron microscopy map of olfactory neurons in drosophila melanogaster for odor classification. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4504–4509. IEEE, 2020.

[49] Jacob Morra and Mark Daley. Using Connectome Features to Constrain Echo State Networks. *arXiv preprint arXiv:2206.02094*, 2022.

[50] Jacob Morra and Mark Daley. Imposing Connectome-Derived Topology on an Echo State Network. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2022.

[51] Lachezar Bozhkov, Petia Koprinkova-Hristova, and Petia Georgieva. Learning to decode human emotions with echo state networks. *Neural Networks*, 78:112–119, 2016.

[52] David Sussillo, Paul Nuyujukian, Joline M Fan, Jonathan C Kao, Sergey D Stavisky, Stephen Ryu, and Krishna Shenoy. A recurrent neural network for closed-loop intracortical brain–machine interface decoders. *Journal of neural engineering*, 9(2):026027, 2012.

[53] Tim Waegeman, Francis wyffels, and Benjamin Schrauwen. Feedback control by online learning an inverse model. *IEEE transactions on neural networks and learning systems*, 23(10):1637–1648, 2012.

[54] Daniel Canaday, Andrew Pomerance, and Daniel J Gauthier. Model-free control of dynamical systems with deep reservoir computing. *Journal of Physics: Complexity*, 2(3):035025, 2021.

[55] Soon Hoe Lim, Ludovico Theo Giorgini, Woosok Moon, and John S Wettlaufer. Predicting critical transitions in multiscale dynamical systems using reservoir computing. *Chaos*, 30(12):123126, 2020.

[56] Ling-Wei Kong, Hua-Wei Fan, Celso Grebogi, and Ying-Cheng Lai. Machine learning prediction of critical transition and system collapse. *Physical Review Research*, 3(1):013090, 2021.

[57] Stuart E Lacy, Stephen L Smith, and Michael A Lones. Using echo state networks for classification: A case study in Parkinson's disease diagnosis. *Artificial intelligence in medicine*, 86:53–59, 2018.

[58] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[59] Zhixin Lu, Brian R Hunt, and Edward Ott. Attractor reconstruction by machine learning. *Chaos*, 28(6):061104, 2018.

[60] Azarakhsh Jalalvand, Glenn Van Wallendael, and Rik Van de Walle. Real-time reservoir computing network-based systems for detection tasks on visual contents. In *2015 7th International Conference on Computational Intelligence, Communication Systems and Networks*, pages 146–151. IEEE, 2015.

[61] Anıl Özdemir, Mark Scerri, Andrew B Barron, Andrew Philippides, Michael Mangan, Eleni Vasilaki, and Luca Manneschi. Echovpr: Echo state networks for visual place recognition. *IEEE Robotics and Automation Letters*, 7(2):4520–4527, 2022.

[62] Pieter Buteneers, David Verstraeten, Bregt Van Nieuwenhuyse, Dirk Stroobandt, Robrecht Raedt, Kristl Vonck, Paul Boon, and Benjamin Schrauwen. Real-time detection of epileptic seizures in animal models using reservoir computing. *Epilepsy research*, 103(2-3):124–134, 2013.

[63] Zhixin Lu, Jaideep Pathak, Brian Hunt, Michelle Girvan, Roger Brockett, and Edward Ott. Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos*, 27(4):041102, 2017.

[64] Jaideep Pathak, Zhixin Lu, Brian R. Hunt, Michelle Girvan, and Edward Ott. Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos*, 27(12):121102, 2017.

[65] Amitava Banerjee, Jaideep Pathak, Rajarshi Roy, Juan G Restrepo, and Edward Ott. Using machine learning to assess short term causal dependence and infer network links. *Chaos*, 29(12):121104, 2019.

[66] Steven H Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering.* CRC Press, 2018.

[67] Arkady Pikovsky, Michael Rosenblum, and Jürgen Kurths. *Synchronization: A universal concept in nonlinear sciences.* Cambridge University Press, 2001.

[68] Stefano Boccaletti, Alexander N Pisarchik, Charo I Del Genio, and Andreas Amann. *Synchronization: from coupled systems to complex networks*. Cambridge University Press, 2018.

[69] Louis M Pecora and Thomas L Carroll. Synchronization in chaotic systems. *Physical Review Letters*, 64(8):821, 1990.

[70] Nikolai F Rulkov, Mikhail M Sushchik, Lev S Tsimring, and Henry DI Abarbanel. Generalized synchronization of chaos in directionally coupled chaotic systems. *Physical Review E*, 51(2):980, 1995.

[71] Ljupco Kocarev and Ulrich Parlitz. Generalized synchronization, predictability, and equivalence of unidirectionally coupled dynamical systems. *Physical Review Letters*, 76(11):1816, 1996.

[72] K Pyragas. Properties of generalized synchronization of chaos. *Nonlinear Analysis: Modelling and Control*, 3:101–129, 1998.

[73] Henry DI Abarbanel, Nikolai F Rulkov, and Mikhail M Sushchik. Generalized synchronization of chaos: The auxiliary system approach. *Physical Review E*, 53(5):4528, 1996.

[74] Jason A Platt, Adrian Wong, Randall Clark, Stephen G Penny, and Henry DI Abarbanel. Robust forecasting using predictive generalized synchronization in reservoir computing. *Chaos*, 31(12):123118, 2021.

[75] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.

[76] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, 120(2):024102, 2018.

[77] Joschka Herteux and Christoph Räth. Breaking symmetries of the reservoir equations in echo state networks. *Chaos*, 30(12):123142, 2020.

[78] Junjie Jiang and Ying-Cheng Lai. Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius. *Physical Review Research*, 1(3):033056, 2019.

[79] Luca Anthony Thiede and Ulrich Parlitz. Gradient based hyperparameter optimization in echo state networks. *Neural Networks*, 115:23–29, 2019.

[80] Jan Yperman and Thijs Becker. Bayesian optimization of hyper-parameters in reservoir computing. *arXiv preprint arXiv:1611.05193*, 2016.

[81] Daniel J Gauthier, Erik Bollt, Aaron Griffith, and Wendson AS Barbosa. Next Generation Reservoir Computing. *Nature Communications*, 12(1), 2021.

[82] Erik Bollt. On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD. *Chaos*, 31(1):013108, 2021.

[83] Lennert Appeltant, Miguel Cornelles Soriano, Guy Van der Sande, Jan Danckaert, Serge Massar, Joni Dambre, Benjamin Schrauwen, Claudio R Mirasso, and Ingo Fischer. Information processing using a single dynamical node as complex system. *Nature communications*, 2(1):1–6, 2011.

[84] Claudio Gallicchio, Alessio Micheli, and Luca Pedrelli. Deep reservoir computing: a critical experimental analysis. *Neurocomputing*, 268:87–99, 2017.

[85] Mirko Goldmann, Felix Köster, Kathy Lüdge, and Serhiy Yanchuk. Deep time-delay reservoir computing: Dynamics and memory capacity. *Chaos*, 30(9):093124, 2020.

[86] Thomas L Carroll and Louis M Pecora. Network structure effects in reservoir computers. *Chaos*, 29(8):083130, 2019.

[87] Jaideep Pathak, Alexander Wikner, Rebeckah Fussell, Sarthak Chandra, Brian R Hunt, Michelle Girvan, and Edward Ott. Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. *Chaos*, 28(4):041101, 2018.

[88] Alexander Wikner, Jaideep Pathak, Brian Hunt, Michelle Girvan, Troy Arcomano, Istvan Szunyogh, Andrew Pomerance, and Edward Ott. Combining machine learning with knowledge-based modeling for scalable forecasting and subgrid-scale closure of large, complex, spatiotemporal systems. *Chaos*, 30(5):053111, 2020.

[89] Troy Arcomano, Istvan Szunyogh, Alexander Wikner, Jaideep Pathak, Brian R Hunt, and Edward Ott. A hybrid approach to atmospheric modeling that combines machine learning with a physics-based numerical model. *Journal of Advances in Modeling Earth Systems*, 14(3):e2021MS002712, 2022.

[90] Laurent Larger, Miguel C Soriano, Daniel Brunner, Lennert Appeltant, Jose M Gutiérrez, Luis Pesquera, Claudio R Mirasso, and Ingo Fischer. Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Optics express*, 20(3):3241–3249, 2012.

[91] Benjamin Cramer, David Stöckel, Markus Kreft, Michael Wibral, Johannes Schemmel, Karlheinz Meier, and Viola Priesemann. Control of criticality and computation in spiking neuromorphic networks with plasticity. *Nature Communications*, 11(1):1–11, 2020.

[92] Richard W Dawidek, Thomas J Hayward, Ian T Vidamour, Thomas J Broomhall, Guru Venkat, Mohanad Al Mamoori, Aidan Mullen, Stephan J Kyle, Paul W Fry, Nina-Juliane Steinke, et al. Dynamically driven emergence in a nanomagnetic system. *Advanced Functional Materials*, 31(15):2008389, 2021.

[93] Karolos-Alexandros Tsakalos, Georgios Ch Sirakoulis, Andrew Adamatzky, and Jim Smith. Protein structured reservoir computing

for spike-based pattern recognition. *IEEE Transactions on Parallel and Distributed Systems*, 33(2):322–331, 2021.

[94] Dawid Przyczyna, Maciej Suchecki, Andrew Adamatzky, and Konrad Szaciłowski. Towards embedded computation with building materials. *Materials*, 14(7):1724, 2021.

[95] Kohei Nakajima, Helmut Hauser, Tao Li, and Rolf Pfeifer. Information processing via physical soft body. *Scientific reports*, 5:10487, 2015.

[96] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, 2019.

[97] Kohei Nakajima. Physical reservoir computing—an introductory perspective. *Japanese Journal of Applied Physics*, 59(6):060501, 2020.

[98] Matthew Dale, Julian F Miller, and Susan Stepney. Reservoir computing as a model for in-materio computing. In *Advances in Unconventional Computing*, pages 533–571. Springer, 2017.

[99] Susan Stepney. Computing with open dynamical systems. In *2021 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, pages 139–143. IEEE, 2021.

[100] Kohei Nakajima and Ingo Fischer. *Reservoir Computing*. Springer, 2021.

[101] Ling-Wei Kong, Yang Weng, Bryan Glaz, Mulugeta Haile, and Ying-Cheng Lai. Digital twins of nonlinear dynamical systems. *arXiv preprint arXiv:2210.06144*, 2022.

[102] André Röhm, Daniel J Gauthier, and Ingo Fischer. Model-free inference of unseen attractors: Reconstructing phase space features from a single noisy trajectory using reservoir computing. *Chaos*, 31(10):103127, 2021.

[103] Kevin L Briggman and William B Kristan. Imaging dedicated and multifunctional neural circuits generating distinct behaviors. *Journal of Neuroscience*, 26(42):10925–10933, 2006.

[104] Ion R Popescu and William N Frost. Highly dissimilar behaviors mediated by a multifunctional network in the marine mollusk tritonia diomedea. *Journal of Neuroscience*, 22(5):1985–1993, 2002.

[105] Ulrike Feudel. Complex dynamics in multistable systems. *International Journal of Bifurcation and Chaos*, 18(06):1607–1626, 2008.

[106] Mikhail I Rabinovich, Pablo Varona, Allen I Selverston, and Henry DI Abarbanel. Dynamical principles in neuroscience. *Reviews of modern physics*, 78(4):1213, 2006.

[107] Eugene M Izhikevich. *Dynamical Systems in Neuroscience*. MIT press, 2007.

[108] G Bard Ermentrout and David H Terman. *Mathematical Foundations of Neuroscience*, volume 35. Springer, 2010.

[109] Daniel M Wolpert and Mitsuo Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7-8):1317–1329, 1998.

[110] Jun Tani, Masato Ito, and Yuuya Sugita. Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using rnnpb. *Neural Networks*, 17(8-9):1273–1289, 2004.

[111] Herbert Jaeger. Using conceptors to manage neural long-term memories for temporal patterns. *The Journal of Machine Learning Research*, 18(1):387–429, 2017.

[112] Katsuma Inoue, Kohei Nakajima, and Yasuo Kuniyoshi. Designing spontaneous behavioral switching via chaotic itinerancy. *Science advances*, 6(46):eabb3989, 2020.

[113] Zhixin Lu and Danielle S Bassett. Invertible generalized synchronization: A putative mechanism for implicit learning in neural systems. *Chaos*, 30(6):063133, 2020.

[114] Jason Z Kim, Zhixin Lu, Erfan Nozari, George J Pappas, and Danielle S Bassett. Teaching recurrent neural networks to infer global temporal structure from local examples. *Nature Machine Intelligence*, 3(4):316–323, 2021.

[115] Ling-Wei Kong, Huawei Fan, Celso Grebogi, and Ying-Cheng Lai. Emergence of transient chaos and intermittency in machine learning. *Journal of Physics: Complexity*, 2(3):035014, 2021.

[116] Huawei Fan, Ling-Wei Kong, Ying-Cheng Lai, and Xingang Wang. Anticipating synchronization with machine learning. *Physical Review Research*, 3(2):023237, 2021.

[117] Mohamed Oubbati and Günther Palm. A neural framework for adaptive robot control. *Neural Computing and Applications*, 19(1):103–114, 2010.

[118] Mohamed Oubbati, Paul Levi, and Michael Schanz. Meta-learning for adaptive identification of non-linear dynamical systems. In *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pages 473–478. IEEE, 2005.

[119] Mohamed Oubbati, Paul Levi, and Michael Schanz. A fixed-weight rnn dynamic controller for multiple mobile robots. In *Proceedings of 24th IASTED international conference on modelling, identification, and control, Innsbruck, Austria*, pages 277–282, 2005.

[120] Lee A Feldkamp, Gintaras V Puskorius, and PC Moore. Adaptation from fixed weight dynamic networks. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pages 155–160. IEEE, 1996.

[121] Danil V Prokhorov, LA Feldkamp, and I Yu Tyukin. Adaptive behavior with fixed weights in rnn: an overview. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, volume 3, pages 2018–2022. IEEE, 2002.

[122] Roberto A Santiago. Context discerning multifunction networks: Reformulating fixed weight neural networks. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, volume 1, pages 189–194. IEEE, 2004.

[123] Andrea Ceni, Peter Ashwin, and Lorenzo Livi. Interpreting recurrent neural networks behaviour via excitable network attractors. *Cognitive Computation*, 12(2):330–356, 2020.

[124] Alexander N Pisarchik and Ulrike Feudel. Control of multistability. *Physics Reports*, 540(4):167–218, 2014.

[125] Ulrike Feudel, Alexander N Pisarchik, and Kenneth Showalter. Multistability and tipping: From mathematics and physics to climate and brain—minireview and preface to the focus issue. *Chaos*, 28(3):033501, 2018.

[126] Omri Barak. Recurrent neural networks as versatile tools of neuroscience research. *Current opinion in neurobiology*, 46:1–6, 2017.

[127] Ronald M Harris-Warrick and Eve Marder. Modulation of neural networks for behavior. *Annual review of neuroscience*, 14(1):39–57, 1991.

[128] Paul E O' Keeffe and Sebastian Wieczorek. Tipping phenomena and points of no return in ecosystems: beyond classical bifurcations. *SIAM Journal on Applied Dynamical Systems*, 19(4):2371–2402, 2020.

[129] Ichiro Tsuda. Chaotic itinerancy and its roles in cognitive neurodynamics. *Current opinion in neurobiology*, 31:67–71, 2015.

[130] Zhi-Hong Guan, Qiang Lai, Ming Chi, Xin-Ming Cheng, and Feng Liu. Analysis of a new three-dimensional system with multiple chaotic attractors. *Nonlinear Dynamics*, 75(1-2):331–343, 2014.

[131] Hongtao Lu. Chaotic attractors in delayed neural networks. *Physics Letters A*, 298(2-3):109–116, 2002.

[132] Bocheng Bao, Hui Qian, Jiang Wang, Quan Xu, Mo Chen, Huagan Wu, and Yajuan Yu. Numerical analyses and experimental validations of coexisting multiple attractors in hopfield neural network. *Nonlinear Dynamics*, 90(4):2359–2369, 2017.

[133] Hendrik Richter. Controlling chaotic systems with multiple strange attractors. *Physics Letters A*, 300(2-3):182–188, 2002.

[134] Eusebius J. Doedel, Thomas F. Fairgrieve, Björn Sandstede, Alan R. Champneys, Yuri A. Kuznetsov, and Xianjun Wang. Auto-07p: Continuation and bifurcation software for ordinary differential equations, 2007.

[135] Yuri A Kuznetsov. *Elements of applied bifurcation theory*, volume 112. Springer Science & Business Media, 2013.

[136] John Guckenheimer and Philip Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42. Springer Science & Business Media, 2013.

[137] EJ Doedel, AR Champneys, F Dercole, TF Fairgrieve, YA Kuznetsov, B Oldeman, RC Paffenroth, B Sandstede, XJ Wang, and CH Zhang. Auto-07p: Continuation and bifurcation software for ordinary differential equations, 2007.

[138] P Tass, MG Rosenblum, J Weule, J Kurths, A Pikovsky, J Volkmann, A Schnitzler, and H-J Freund. Detection of n: m phase locking from noisy data: Application to magnetoencephalography. *Physical Review Letters*, 81(15):3291, 1998.

[139] Michael Rosenblum and Arkady Pikovsky. Delayed feedback control of collective synchrony: An approach to suppression of pathological brain rhythms. *Physical Review E*, 70(4):041904, 2004.

[140] Peter A Tass, Christian Hauptmann, and Oleksandr V Popovych. Development of therapeutic brain stimulation techniques with methods from nonlinear dynamics and statistical physics. *International Journal of Bifurcation and Chaos*, 16(07):1889–1911, 2006.

[141] Thomas Lymburn, Shannon D Algar, Michael Small, and Thomas Jüngling. Reservoir computing with swarms. *Chaos*, 31(3):033121, 2021.

[142] Wendson AS Barbosa, Aaron Griffith, Graham E Rowlands, Luke CG Govia, Guilhem J Ribeill, Minh-Hai Nguyen, Thomas A Ohki, and Daniel J Gauthier. Symmetry-aware reservoir computing. *Physical Review E*, 104(4):045307, 2021.

[143] Nils Bertschinger and Thomas Natschläger. Real-time computation at the edge of chaos in recurrent neural networks. *Neural computation*, 16(7):1413–1436, 2004.

[144] Joschka Boedecker, Oliver Obst, Joseph T Lizier, N Michael Mayer, and Minoru Asada. Information processing in echo state networks at the edge of chaos. *Theory in Biosciences*, 131(3):205–213, 2012.

[145] Christof Teuscher. Revisiting the edge of chaos: Again? *Biosystems*, page 104693, 2022.

[146] Celso Grebogi, Edward Ott, and James A Yorke. Fractal basin boundaries, long-lived chaotic transients, and unstable-unstable pair bifurcation. *Physical Review Letters*, 50(13):935, 1983.

[147] Celso Grebogi, Steven W McDonald, Edward Ott, and James A Yorke. Final state sensitivity: an obstruction to predictability. *Physics Letters A*, 99(9):415–418, 1983.

[148] Steven W McDonald, Celso Grebogi, Edward Ott, and James A Yorke. Fractal basin boundaries. *Physica D: Nonlinear Phenomena*, 17(2):125–153, 1985.

[149] Herbert Jaeger. Short term memory in echo state networks. gmd-report 152. In *GMD-German National Research Institute for Computer Science*. Citeseer, 2002.

[150] Thomas L Carroll and Joseph D Hart. Time shifts to reduce the size of reservoir computers. *Chaos*, 32(8):083122, 2022.

[151] Natalia B Janson and Christopher J Marsden. Dynamical system with plastic self-organized velocity field as an alternative conceptual model of a cognitive system. *Scientific Reports*, 7(1):1–15, 2017.

[152] Jacques Bélair, Fahima Nekka, and John G Milton. Introduction to focus issue: Dynamical disease: A translational approach, 2021.

[153] Cian McCafferty, Francois David, Marcello Venzi, Magor L Lőrincz, Francis Delicata, Zoe Atherton, Gregorio Recchia, Gergely Orban, Régis C Lambert, Giuseppe Di Giovanni, Nathalie Leresche, and Vincenzo Crunelli. Cortical drive and thalamic feed-forward inhibition control thalamic output synchrony during absence seizures. *Nature neuroscience*, 21(5):744–756, 2018.

[154] Fernando Lopes da Silva, Wouter Blanes, Stiliyan N Kalitzin, Jaime Parra, Piotr Suffczynski, and Demetrios N Velis. Epilepsies as dynamical diseases of brain systems: basic models of the transition between normal and epileptic activity. *Epilepsia*, 44:72–83, 2003.

[155] Fernando H Lopes da Silva, Wouter Blanes, Stiliyan N Kalitzin, Jaime Parra, Piotr Suffczynski, and Demetrios N Velis. Dynamical diseases of brain systems: different routes to epileptic seizures. *IEEE transactions on biomedical engineering*, 50(5):540–548, 2003.

[156] Piotr Suffczynski, Stiliyan Kalitzin, and FH Lopes da Silva. Dynamics of non-convulsive epileptic phenomena modeled by a bistable neuronal network. *Neuroscience*, 126(2):467–484, 2004.

[157] Piotr Suffczynski, Fernando Lopes da Silva, Jaime Parra, Demetrios Velis, and Stiliyan Kalitzin. Epileptic transitions: model predictions and

experimental validation. *Journal of clinical neurophysiology*, 22(5):288–299, 2005.

[158] FH Lopes da Silva, A Hoeks, H Smits, and LH Zetterberg. Model of brain rhythmic activity. the alpha-rhythm of the thalamus. *Kybernetik*, 15(1):27–37, 1974.

[159] Michael Breakspear, James A Roberts, John R Terry, Serafim Rodrigues, Neil Mahant, and Peter A Robinson. A unifying explanation of primary generalized seizures through nonlinear brain modeling and bifurcation analysis. *Cerebral Cortex*, 16(9):1296–1313, 2006.

[160] PA Robinson, CJ Rennie, and DL Rowe. Dynamics of large-scale brain activity in normal arousal states and epileptic seizures. *Physical Review E*, 65(4):041924, 2002.

[161] Daisuke Takeshita, Yasuomi D Sato, and Sonya Bahar. Transitions between multistable states as a model of epileptic seizure dynamics. *Physical Review E*, 75(5):051925, 2007.

[162] Hugh R Wilson. Simplified dynamics of human and mammalian neocortical neurons. *Journal of theoretical biology*, 200(4):375–388, 1999.

[163] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.

[164] Frank Marten, Serafim Rodrigues, Piotr Suffczynski, Mark P Richardson, and John R Terry. Derivation and analysis of an ordinary differential equation mean-field model for studying clinically recorded epilepsy dynamics. *Physical Review E*, 79(2):021911, 2009.

[165] Frank Marten, Serafim Rodrigues, Oscar Benjamin, Mark P Richardson, and John R Terry. Onset of polyspike complexes in a mean-field model of human electroencephalography and its application to absence epilepsy. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1891):1145–1161, 2009.

[166] Marc Goodfellow, Kaspar Schindler, and Gerold Baier. Intermittent spike–wave dynamics in a heterogeneous, spatially extended neural mass model. *NeuroImage*, 55(3):920–932, 2011.

[167] Ben H Jansen, George Zouridakis, and Michael E Brandt. A neurophysiologically-based mathematical model of flash visual evoked potentials. *Biological cybernetics*, 68(3):275–283, 1993.

[168] Ben H Jansen and Vincent G Rit. Electroencephalogram and visual evoked potential generation in a mathematical model of coupled cortical columns. *Biological cybernetics*, 73(4):357–366, 1995.

[169] JL Perez Velazquez, Houman Khosravani, Andres Lozano, Berj L Bardakjian, Peter L Carlen, and Richard Wennberg. Type III intermittency in human partial epilepsy. *European Journal of Neuroscience*, 11(7):2571–2576, 1999.

[170] Alexander Hramov, Alexey A Koronovskii, IS Midzyanovskaya, E Sitnikova, and CM Van Rijn. On-off intermittency in time series of spontaneous paroxysmal activity in rats with genetic absence epilepsy. *Chaos*, 16(4):043111, 2006.

[171] Marc Goodfellow and Paul Glendinning. Mechanisms of intermittent state transitions in a coupled heterogeneous oscillator model of epilepsy. *The Journal of Mathematical Neuroscience*, 3(1):1–14, 2013.

[172] Evgenia Sitnikova, Alexander E Hramov, Vadim V Grubov, Alexey A Ovchinnkov, and Alexey A Koronovsky. On–off intermittency of thalamo-cortical oscillations in the electroencephalogram of rats with genetic predisposition to absence epilepsy. *Brain research*, 1436:147–156, 2012.

[173] Alexey A Koronovskii, Alexander E Hramov, Vadim V Grubov, Olga I Moskalenko, Evgenia Sitnikova, and Alexey N Pavlov. Coexistence of intermittencies in the neuronal network of the epileptic brain. *Physical Review E*, 93(3):032220, 2016.

[174] Gouhei Tanaka, Tadayoshi Matsumori, Hiroaki Yoshida, and Kazuyuki Aihara. Reservoir computing with diverse timescales for prediction of multiscale dynamics. *Physical Review Research*, 4(3):L032014, 2022.

[175] Joschka Herteux. The Influence of the Activation Function on Reservoir Computers. Master's thesis, Ludwig-Maximilians-Universität München, 2021.

[176] Xu Zhang and Guanrong Chen. Constructing an autonomous system with infinitely many chaotic attractors. *Chaos*, 27(7):071101, 2017.

[177] Kenji Doya. Bifurcations in the learning of recurrent neural networks. In *[Proceedings] 1992 IEEE International Symposium on Circuits and Systems*, volume 6, pages 2777–2780. IEEE, 1992.

[178] Randall D Beer. On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4):469–509, 1995.

[179] Randall D Beer. Codimension-2 parameter space structure of continuous-time recurrent neural networks. *Biological Cybernetics*, 116(4):501–515, 2022.

[180] Sonia Harmand, Jason E Lewis, Craig S. Feibel, and et al. 3.3-million-year-old stone tools from Lomekwi 3, West Turkana, Kenya. *Nature*, 521(7552):310–315, 2015.

[181] PwC. PwC's Global Artificial Intelligence Study: Exploiting the AI Revolution, 2019. `www.pwc.com/gx/en/issues/data-and-analytics/publications/artificial-intelligence-study.html`, Last accessed on 30-01-2023.

[182] Tambiama Madiega. Eu guidelines on ethics in artificial intelligence: Context and implementation. *European Parliamentary Research Service*, 2019.

[183] Ray Kurzweil. *The singularity is near: When humans transcend biology.* Penguin, 2005.

[184] Sean Flynn. Online Talk: William Gibson "Burning Chrome". `https://youtu.be/iaBbSjYoG5M`, Last accessed on 30-01-2023.

[185] James Barrat. *Our final invention: Artificial intelligence and the end of the human era.* Macmillan, 2013.

[186] Slavoj Žižek. *Hegel in a Wired Brain.* Bloomsbury Publishing, 2020.

[187] Mark O'Connell. *To Be a Machine: Adventures Among Cyborgs, Utopians, Hackers, and the Futurists Solving the Modest Problem of Death.* Granta Books, 2017.

[188] Slavoj Žižek. *First as Tragedy, Then as Farce.* Verso, 2009.

[189] Henri Poincaré. *Les méthodes nouvelles de la mécanique céleste: Méthodes de MM. Newcomb, Gyldén, Linstadt et Bohlin*, volume 2. Gauthier-Villars et fils, imprimeurs-libraires, 1893.

[190] Christophe Letellier, Ralph Abraham, Dima L Shepelyansky, Otto E Rössler, Philip Holmes, René Lozi, Leon Glass, Arkady Pikovsky, Lars F Olsen, Ichiro Tsuda, et al. Some elements for a history of the dynamical systems theory. *Chaos*, 31(5):053110, 2021.

[191] Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International journal of control*, 55(3):531–534, 1992.

[192] Gaston Floquet. Sur les équations différentielles linéaires à coefficients périodiques. In *Annales scientifiques de l'École Normale Supérieure*, volume 12, pages 47–88. Elsevier, 1883.