



HAL
open science

Complex signal embedding and photonic reservoir Computing in time series prediction

Bicky Marquez Alfonzo

► **To cite this version:**

Bicky Marquez Alfonzo. Complex signal embedding and photonic reservoir Computing in time series prediction. Neural and Evolutionary Computing [cs.NE]. Université Bourgogne Franche-Comté, 2018. English. NNT : 2018UBFCD042 . tel-02096857

HAL Id: tel-02096857

<https://theses.hal.science/tel-02096857>

Submitted on 11 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



SPIM

Thèse de Doctorat



UFC

école doctorale **sciences pour l'ingénieur et microtechniques**
UNIVERSITÉ DE FRANCHE-COMTÉ

Complex signal embedding and
photonic Reservoir Computing in
time series prediction

■ BICKY A. MARQUEZ

SPIM

Thèse de Doctorat



école doctorale **sciences pour l'ingénieur et microtechniques**
UNIVERSITÉ DE FRANCHE-COMTÉ



THÈSE présentée par
BICKY A. MARQUEZ

pour obtenir le
Grade de Docteur de
l'Université de Bourgogne Franche-Comté

Spécialité : **Optique et Photonique**

Reservoir Computing Photonique et Méthodes Non-Linéaires de Représentation de Signaux Complexes: Application à la Prédiction de Séries Temporelles

Soutenue publiquement le 27 mars 2018 devant le Jury composé de :

ECKEHARD SCHÖLL	Rapporteur	Professeur à Technischen Universität Berlin
STÉPHANE BARLAND	Rapporteur	Chargé de Recherche à Université Côte d'Azur
MAXIME JACQUOT	Directeur de thèse	Maitre de conférences à l'Université Bourgogne - Franche-Comté
LAURENT LARGER	CoDirecteur de thèse	Professeur à l'Institut FEMTO-ST
DANIEL BRUNNER	CoDirecteur de thèse	Chargé de Recherche à l'Institut FEMTO-ST
LYUDMILA GRIGORYEVA	Examinateur	Professeur assistant à l'Universität Konstanz
LAURENT DAUDET	Examinateur	Professeur à l'Université Pierre et Marie Curie
MARIA-PILAR BERNAL	Examinateur	Directeur de Recherche à l'Institut FEMTO-ST

Acronyms and abbreviations list

A

ANN: artificial neural network.

ARMA: autorregressive-moving-average.

AC: autocorrelation analysis.

Amp: RF amplifier.

B

BPF: bandpass filter.

C

CCA: cross-correlation analysis.

D

DDE: delay differential equation.

DL: delay line.

DRC: delay reservoir computing.

E

ESN: echo state network.

F

FNN: false nearest neighbors.

FHN: FitzHugh-Nagumo.

FIFO: fist-in first-out.

I

ISI: interspike interval.

L

LSM: liquid state machine.

LSE: last Square Estimation.

M

MG: Mackey Glass.

MI: mutual information.

MARS: multi Adaptive Regression Spline.

MZM: Mach-Zehnder modulator.

N

NMSE: normalize mean squared error.

NN: nearest neighbors.

NARX: nonlinear Autoregressive with Exogenous inputs.

O

ODE: ordinary differential equation.

P

PPF: proportional perturbation feedback.

PD: Photodiode.

PH: prediction horizon.

PRC: photonic reservoir computer.

R

RNN: recurrent neural network.

rRNN: random recurrent neural network.

RP: random projection.

RC: reservoir computing.

S

SVD: singular value decomposition.

SVM: support Vector Machine.

T

TrRNN: Takens random recurrent neural network.

TRC: Takens RC.

List of Symbols

List of symbols in chronological order of appearance.

Chapter 1: Fundamental concepts

M : function in a discrete time representation.

r : bifurcation parameter of the logistic equation.

F : function in a continuous time representation.

a_1 : number of Prandtl.

b_1 : Rayleigh number.

c_1 : scaling factor in the Lorenz system.

$a_2, b_2,$ and c_2 : effective reaction rates in the Rössler system.

ϑ, ν and ψ : parameters of the Mackey Glass system.

τ_m : time delay of the Mackey Glass system.

R : reflectivity.

η : nonlinear refractive index.

η_2 : quadratic coefficient of the nonlinear refractive index.

l_c : length of the nonlinear dielectric medium in the Ikeda system.

ρ : absorption coefficient.

k_w : wave number.

L : length of the cavity in the Ikeda system.

E : optic field.

E_I : is the amplitude of the incident field.

A : is a parameter proportional to the incident field amplitude.

B : is a parameter that describes the dissipation in the cavity.

ϕ_0 : constant offset.

c : speed of light in vacuum.

Ω : mapping rule.

E_n : iterative variable for the optical field.

I_n : intensity.

ϕ : variable for the relaxation time of the nonlinear medium.

γ : relaxation rate.

t_R : time delay of the Ikeda system.

f_c : cutoff frequency.

β : gain parameter.

f_{NL} : nonlinear function.

τ_D : time delay of the Ikeda-like system.

$h(t)$: impulse response of a filter.

$x(t), y(t)$ and $z(t)$: dynamical variables.

t : temporal variable.

$X(\omega)$, and $Y(\omega)$: dynamical variables in the frequency domain.

τ : response time of the filter.

M : dimension of an original attractor.

τ_0 : lag at the first minimum of the autocorrelation analysis.

R_{yy} : autocorrelation function.

l : lag.

g : size of the dataset.

M_1 : parameter of dimension.

\mathbf{y}_n : state at time n .

$R_{\mathbf{y}_n}$: rule to define false nearest neighbors.

d_l : interstate distance.

λ_{max} : maximum Lyapunov exponent.

Ψ : probability density function.

\mathbf{W} : internal weights.

W^{off} : weights of the bias.

b : bias.

Ξ : error function.

T : prediction horizon.

\mathbf{x}_n : state matrix of the network.

W^{fb} : input/feedback weights.

y_{n+1}^T : input and teacher signal.

y^{out} : output of the network.

W^{out} : output weights.

m : dimension of the network.

σ_k : temporal position of each node.

σ : short time scale of the dynamics.

α amplification parameter.

u_{in} : time multiplexed input.

W^{in} : input weights.

$\delta\tau$: temporal spacing between neighbor nodes.

Λ : symmetric matrix.

V : orthogonal basis in the domain of the mapping.

U : orthogonal basis in the co-domain of the mapping.

Σ : diagonal matrix.

δ : standard deviation.

M_x : node responses matrix.

M_y^T : target matrix.

\mathbf{I}_m : identity matrix.

λ : regression parameter.

Chapter 2: State-space prediction in Random Neural Networks

d, k, q and h : dimensional parameters of random mapping.

ϵ : interpoint distances limit in linear random mapping.

ϵ_1 and ϵ_2 interpoint distances limits in nonlinear random mapping.

L_2 : norm of a vector is the square root of the sum of the absolute values squared.

\mathbb{P} : linear map.

φ : nonlinear map.

\mathbf{x}_n : state matrix of the network.

W^{fb} : input/feedback weights.

y_{n+1}^T : input and teacher signal.

y^{out} : output of the network.

W^{out} : output weights.

W^{off} : weights of the bias.

b : bias.

m : dimension of the network.

μ : bifurcation parameter / spectral radius.

R_{xy} : cross-correlation function.

l : lag.

l_{min}^{MG} and l_{max}^{MG} : minimum and maximum lag limits for the Mackey Glass system.

l_{min}^{IK} and l_{max}^{IK} : minimum and maximum lag limits for the Ikeda system.

τ_0^{MG} and τ_0^{IK} : lag at the first minimum of the autocorrelation analysis for Mackey Glass and Ikeda systems, respectively.

σ : order of uncertainty.

d_o : original interpoint distances.

ξ_n^j : distance.

ω : sampling distribution half-width.

ε_n^j : node-specific error contribution.

$v(t)$ and $\omega(t)$: dynamical variables of the FitzHugh-Nagumo neuronal model.

$\epsilon_{fhn}, g_{fhn}, D_{fhn}$ and h_{fhn} : parameters of the FitzHugh-Nagumo neuronal model.

$\xi(t)$: Gaussian white noise.

τ_\star : position of the fixed point.

τ_i : small collection of interspike intervals.

a_{fhn}, b_{fhn} and c_{fhn} : parameters of the proportional perturbation feedback method.

τ_{max} : control variable.

A_{fhn} and C_{fhn} : parameters of the control equation.

Chapter 3: Dynamical complexity and computation in neural networks beyond their fixed point

\mathbf{x}_n : state matrix of the network.

W^{fb} : input/feedback weights.

y_{n+1}^T : input and teacher signal.

y^{out} : output of the network.

W^{out} : output weights.

W^{off} : weights of the bias.

b : bias.

m : dimension of the network.

μ : bifurcation parameter / spectral radius.

$R_1, R_2, R_3, R_4,$ and R_5 : non-chaotic regimes.

α amplification parameter.

f_{NL} : nonlinear function.

τ_m : time delay of the Mackey Glass system.

σ : signal's standard deviation.

$\sigma_{\alpha=0}$: statistical amplitude variation.

MI_i : mutual information shared between node responses.

Ψ : probability density function.

C_m : memory capacity.

λ_{max} : maximum Lyapunov exponent.

Chapter 4: Electro-optic oscillator as delay neural network

I_0 : laser diode threshold current.

P_0 : power at the output of the Mach-Zhender modulator.

$V(t)$: voltage at the output of the filter.

$V_{\pi RF}$: RF half-wave voltage.

ϕ_0 : bias.

V_{DC} : DC voltage.

τ_D : time delay.

V_{pd} : voltage of the photodiode.

P_{in} : input power.

S : sensitivity of the photodiode.

V_{in} : input voltage to the filter.

f_l : low cutoff frequency.

f_h : high cutoff frequency.

R_1, R_2 and R_3 : resistances to build the bandpass filter.

C_1 and C_2 : capacitances to build the bandpass filter.

G : gain parameter.

β : feedback gain parameter.

τ : fast response time of the filter.

θ : slow response time of the filter.

ζ : dimensionless time.

ϑ and δ : are dimensionless parameters of the bandpass Ikeda.

λ : eigenvalues.

ω : eigen angular frequency.

T_P : period of the fast-scale oscillations.

$F(x), g(x)$ and $G(x)$: variables of the Liénard system.

$z(t)$: dynamical variable to model the interaction between Liénard and Ikeda dynamics.

$z_i, f(\iota)$ and $g(\iota)$: dynamical variables of the space-time-continuous representation of Ikeda system.

ι : dimensionless time.

R : dimensionless time delay.

ρ : dissipative coefficient.

$z_s, f(k)$ and $g(k)$: dynamical variables of the space-time-discrete representation of Ikeda system.

h : integration time.

s : spatial component.

k : temporal component.

M^x : node responses matrix.

T : prediction horizon.

m : dimension of the network.

ϵ_1 and ϵ_2 interpoint distances limits in nonlinear random mapping.

Contents

Acronyms and abbreviations list	v
List of Symbols	xii
Introduction	1
1 Fundamental concepts	5
1.1 Dynamical Systems	5
1.1.1 Discrete and continuous time systems	5
1.1.2 Linear and nonlinear systems	7
1.1.3 Bifurcation diagram	8
1.2 Continuous-time chaotic systems	9
1.2.1 Lorenz and Rössler chaotic attractors	9
1.2.2 Mackey-Glass system	11
1.2.3 Ikeda system	13
1.3 Complexity measure and time series analysis	17
1.3.1 Attractor reconstruction	17
1.3.2 Lyapunov exponent	20
1.3.3 Mutual information	22
1.4 Neural networks	22
1.4.1 Biological and artificial Neurons	23
1.4.2 Artificial neural networks	24
1.4.3 Training	27
1.4.4 Benchmark tasks	29
1.4.5 Reservoir computing	30
1.4.6 Reservoir computing in spatio-temporal networks	31
1.4.7 Delay reservoir computing	32
1.4.8 Regression	34
1.5 Conclusion	35
2 State-space prediction in Random Recurrent Neural Networks	37
2.1 State space reconstruction of chaotic systems	38
2.1.1 Random projections theory	38
2.1.2 Random Neural Networks	40
2.2 Nearest neighbours	44

2.2.1	Nearest neighbours from a rRNN	46
2.3	Takens-inspired rRNN	51
2.4	Application: control an arrhythmic neuronal model	54
2.5	Conclusion	58
3	Dynamical complexity and computation in neural networks beyond their fixed point	59
3.1	Random recurrent neural networks	60
3.2	Mitigating autonomous dynamics by learning	63
3.3	Preservation of information features in destabilized rRNNs	69
3.4	Conclusion	72
4	Electro-optic oscillator as delay neural network	73
4.1	Autonomous electro-optical system	74
4.1.1	Electro-optic oscillator setup and model	74
4.1.2	Dynamics of the autonomous system	77
4.1.3	The Liénard-Ikeda approach	80
4.2	Electro-optic oscillator for information processing	85
4.2.1	Spatio-temporal approach	85
4.2.2	Delay reservoir computing for prediction of chaotic time series	90
4.2.3	Photonic reservoir computing for direct prediction	100
4.2.4	Conclusion	103
5	Conclusion and perspectives	105
	Bibliography	108

Introduction

Uncovering patterns in complex and disordered systems has always been a major challenge for humanity. Such activity can be seen as mere whim and curiosity, related to our constant internal questioning about all that surrounds us. However, the adequate study of complexity, and its potential underlying organisation/structure, is also a necessity for humans as individuals, and for society. The fact that it is not always possible to understand what happens in our environment, leads to focus on the study of all what we cannot control. Since then, we see ourselves as explorers in search of an explanation for everything that merges in our circumstance. Such understanding gives us the freedom to manage our lives within an environment that we now could at least partly predict.

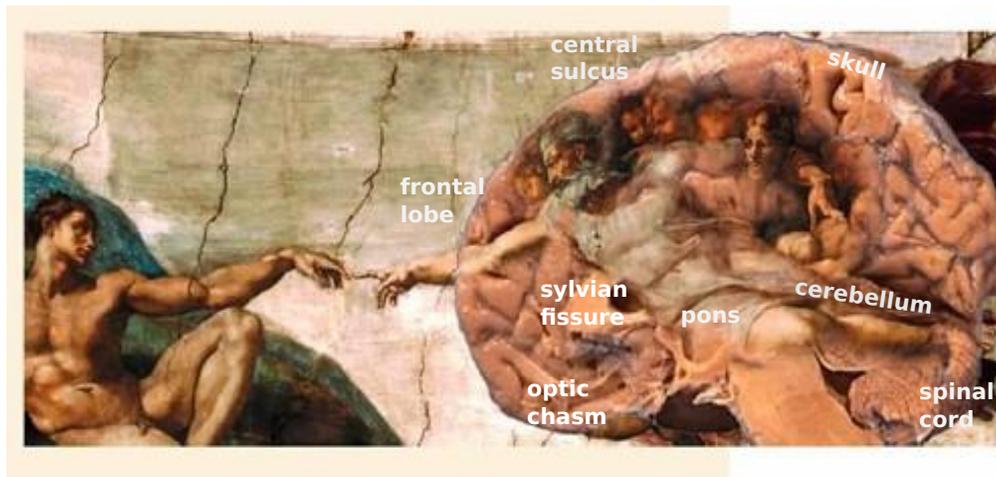


Figure 0.1: An illustrative picture of a section of the Sistine Chapel's Ceiling painted by Michelangelo, which apparently hides the structure of a brain in the god's cloud. The brain was found to be associated to intelligence, and intelligence creates the universe through the hand of god. At this point in creation, the hand of god touches the hand of the man, transmitting then his cognitive capabilities to humans.

Due to our aspirations to model and predict everything, we also include elements that make up our body in the path of knowledge. Among these elements, we highlight the brain as one of the most fascinating organic machine in our body composition, see Fig. 0.1. The human brain contains around 100

billion neurons, which interact with each other to analyse data coming from external stimulus. Its 100 trillion set of synaptic interconnections makes the processing of large amounts of information a task that turns out to be fast and well performed [1]. Therefore, it is natural that at some point we try to construct artificial models that can mimic the brain's capabilities to process data. Nowadays, artificial neural networks do not consist solely of schemes that attempt to reconstruct the anatomy of the brain, they are also used to solve highly complex tasks which often cannot be solved in other known ways (such as digital computing machine). Such tasks can be of different kinds, but the most common and challenging are features classification and prediction of events.

Despite all the sophisticated research about brains (not only human) that has been done, its global functioning remains an enigma for the scientific community. This has led to the creation of many different theoretical models of networks, which solve different problems as the brain is suspected to do. The most important branches containing these models are feedforward and recurrent artificial neural networks [2, 3]. As for their biological counterparts, artificial networks might be trained if we intent to use them to help solving problems. Generally, networks are trained by adjusting their synaptic connections according to a learning rule that can be either supervised by a teacher, or unsupervised. In literature, there exist several methods to train networks depending on their architecture, such as gradient descent, backpropagation, etc [2].

Some recurrent models can be compared with nervous systems' networks of advanced biological species [4]. In fact, recurrent networks can be thought as models where information is recycled, transformed and reused. That is to say, the recurrent property of the network yields circulation of information, which causes the generation of internal dynamics. Consequently, recurrent neural networks are dynamical systems, and they can be studied under the principles of dynamical systems theory [2]. Within this framework, we begin with a description of the tools that are typically used in the study of dynamical systems in general. The first question to ask ourselves is how we can mathematically model a neural network, if it can either be described with iterative maps or differential equations. Then we have to find out if we can find an analytical solution to these equations. If not, we must introduce qualitative methods to analyse their properties, and understand what behavior they may posses according to their architecture.

In our work, we analyse the properties of recurrent networks framed in the concept of Reservoir Computing. This concept first appeared in the early 2000s

with the independent works of H. Jaeger [5] and W. Maass [6], providing an universal framework to achieve high performance computing by using recurrent neural networks with a novel training paradigm. The exciting aspects about this innovative scheme is that it can be easily experimentally implemented in analog hardware, without the need of an excessive amount of resources [7, 8]. This situation places us in the advantage of being able to create replicas of certain brain functions in physical systems, which up to now essentially existed based on numerical algorithms executed on conventional digital computers.

Thesis plan

This PhD thesis is divided into four chapters, involving all work performed within the Optics Department of the FEMTO-ST Institute.

Chapter one provides most of the content related to dynamical systems theory, necessary for the understanding of the manuscript. We develop a conceptual toolbox to analyse all required types of dynamical systems, with a special focus on the analysis of nonlinear phenomena. We introduce chaos theory and show the most representative examples that characterize it. Afterwards, a set of complexity indices and qualitative methods are listed in order to investigate hidden properties of nonlinear systems without analytic solutions. As the final part, artificial neural network theory is introduced. Here, we can find description of biological and artificial neurons, types of artificial networks in the literature, and the introduction of recurrent networks involved in the Reservoir Computing paradigm, which have been exploited in our work from the rather unusual viewpoint of nonlinear dynamics.

In Chapter two, we investigate the procedure that random recurrent neural networks utilize to predict chaotic time series. We describe how networks in steady state employ a version of classical state-space reconstruction techniques well known for chaotic signals. We use the knowledge provided by prediction theory to explain how recurrent networks harness state-space reconstruction to solve prediction. The use of the concept of nearest neighbours proves to be critical for the understanding and realizability of long-term prediction. This is followed by an explanation of the learning role on the output weights, which actually appears to exploit state-space reconstruction capability in the network. This capability allows the network to extract at the output the most accurate prediction from selected internal network states. Finally, inspired by this finding, we present a modified version of the classical random recurrent neural network, in order to build a predictor capable to realize the sufficient

dimensions to fully reconstruct state-spaces of chaotic systems. As a plus, an application of this last scheme is implemented as a control system to stabilize the noisy firing behavior in the FitzHugh-Nagumo neuronal model.

Chapter three contains a study of dynamical complexity in neural networks beyond their fixed point. That is to say, in the previous chapter we were considering networks in steady state. Now, instead of this, let us imagine that our networks can generate periodic, or even multi-periodic dynamics as well. Then, the main aim in this chapter is to investigate how prediction can be solved with this variety of dynamics used as background. The dynamical complexity is measured by means of the spatial synchronization between neurons, memory capacity and maximal Lyapunov exponent of the network.

Chapter four introduces the requirements to implement a recurrent network in an real world analog experimental setup. This network differs from the previous ones in its architecture. We use a recurrent delay network for prediction, inspired by a well-known delay system consisting on a nonlinear optical ring cavity. The proposed experimental system is studied in a nonlinear dynamics framework. Next, we use this delay dynamical system in steady state to implement a neuromorphic processor dedicated to prediction task. Finally, we analytically illustrate how the delay system can be compared to a spatial neural network for prediction.

This manuscript finally provides conclusions and perspectives on the presented research. Our findings will be summarized to suggest new orientations in the analysis of neural networks based on dynamical systems theory, as well as new generation of photonic neuromorphic components.

Chapter 1

Fundamental concepts

1.1 Dynamical Systems

Dynamical systems theory comprises the study and analysis of systems with temporal evolution. The formal study of dynamical models starts with mathematical methods also found in Newtonian mechanics, such as maps and differential equations. Such methods form a toolbox to model the behavior, interaction and changes of the systems under study. As an illustrative example, let us introduce the event of a ball moving on a surface with some friction. If we know all variables involved in this movement (gravitational acceleration, friction coefficient, mass, etc), it is possible to completely describe their trajectories and estimate the final location of the ball. It seems that we can predict what could happen to the ball, depending on our knowledge of its variables. Such cases, where we know all variables and their interaction rules that determine the trajectory, are known as deterministic models. This determinism is characterized by the possibility to determine future states of a given trajectory from the exact knowledge of the present state, through a prescribed causal rule.

However, dynamical systems theory is not exclusively developed to model classical physical systems, such as the example of the ball. It also includes the study of populations dynamics, where the constitution of populations can be investigated through the evolution of density and age. Such populations can be seen as dynamical variables since they change in time. The temporal evolution of many of these models can be defined in a discrete-time representation, where their dynamical variables change at discrete instances [9, 10, 11].

1.1.1 Discrete and continuous time systems

A discrete-time dynamical system is defined mathematically with difference equations or iterative maps, in which the temporal variable n takes values from the set of natural numbers, $n \in \mathbb{N}$,

$$\mathbf{x}_{n+1} = \mathbf{M}(\mathbf{x}_n). \tag{1.1}$$

The variable \mathbf{x}_n provides information about the present state of the system, \mathbf{x}_{n+1} represents the future state to be determined through the causal rule \mathbf{M} .

As previously commented, a discrete-time representation allows us to describe different phenomena that can be similar to population dynamics. In this example, the temporal evolution of the population density is described in generations, and the changes between past and present generations should be represented by the discrete variables \mathbf{x}_{n-1} and \mathbf{x}_n , respectively. Hence, the growth of the population under study could be assumed to always increase in time. Then the function \mathbf{M} is a constant value in Eq. (1.1). However, the population density growth could also perform a more complex evolution path than just monotonous increments. One of the ways to mathematically describe non-monotonous density growth phenomena is via the logistic map [9],

$$\mathbf{x}_{n+1} = r\mathbf{x}_n(1 - \mathbf{x}_n), \quad (1.2)$$

where r defines the population growth rate. In this case, \mathbf{M} is a polynomial function which provides a hypothetical variant feature in the population growth model.

As population growth can be investigated as well via a continuous reproduction activity in some species, continuous changes in the population density may be expected. The logistic equation therefore can be generalized into a continuous-time representation,

$$\frac{d\mathbf{x}}{dt} = r\mathbf{x}(1 - \mathbf{x}). \quad (1.3)$$

This representation serves to model the evolution of variables which have infinitesimal changes in time. Then, the temporal variable t takes values from the set of real numbers, $t \in \mathbb{R}$. In general, continuous-time dynamical systems are described by differential equations,

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}), \quad (1.4)$$

where \mathbf{F} has the role of the function \mathbf{M} in Eq. (1.1).

In both cases, all dynamical variables can interact with each other through constant or polynomial causal rules \mathbf{M} and \mathbf{F} . These prescriptions are parts of more general classes of functions defined as linear and nonlinear functions. The presence of linear or nonlinear components categorize all dynamical models into linear or nonlinear systems.

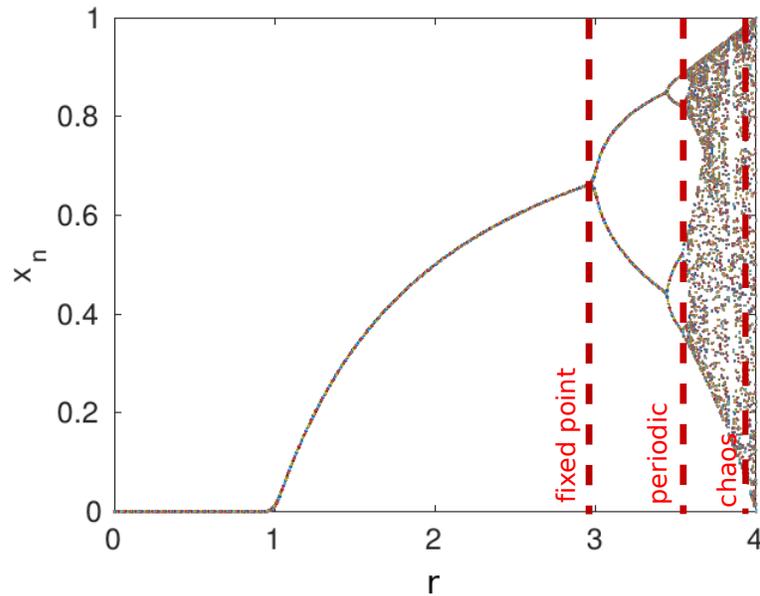


Figure 1.1: Bifurcation diagram of the logistic map.

1.1.2 Linear and nonlinear systems

Let us at first define a generic function f as the equivalent of \mathbf{M} in discrete-time and \mathbf{F} in continuous-time representations. If we define linear systems as the ones which satisfy a superposition principle with respect to the operation of addition, then they can be characterized as linear superposable systems [12]. Hence, given a function f in a system of equations, the additivity property has to be fulfilled:

$$f(x + y) = f(x) + f(y).$$

Thus, linear systems are defined by equations where there are not multiplication between variables, i.e. their dynamical variables appear to the first power only [10]. On the other hand, nonlinear systems can be composed by variables which appear to a power different to one. Multiplication between different variables can also be expected.

The presence of nonlinear components in a dynamical system can lead to the existence of multiple classes of solutions for maps or differential equations. For example, in the logistic map, Eq. (1.2), those different classes of solutions can be found under the variation of parameter r .

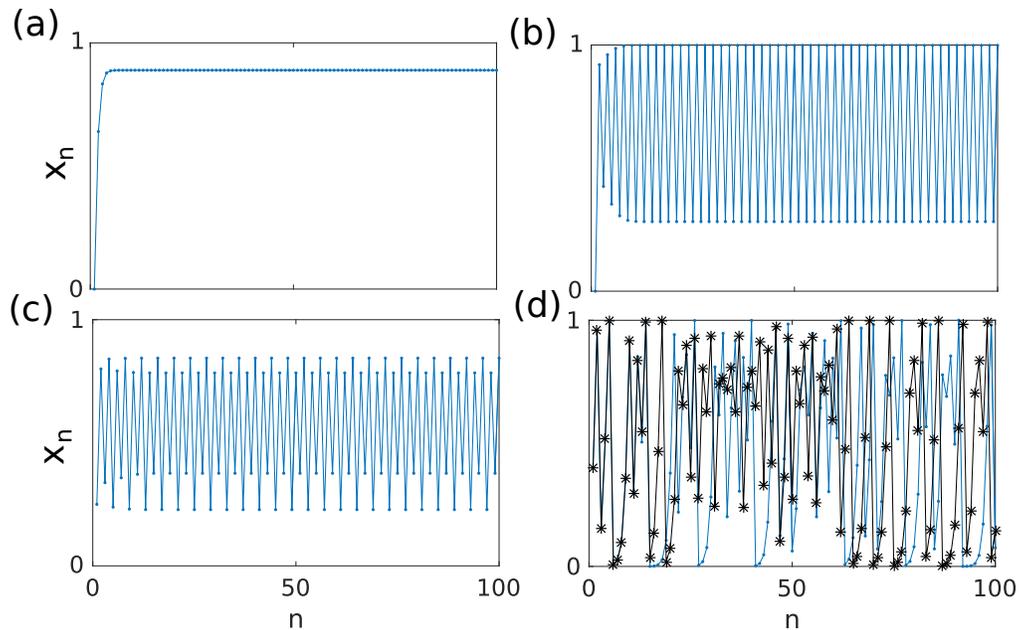


Figure 1.2: Solutions of the logistic map for (a) $r = 0.5$, (b) $r = 1$, (c) $r = 2.5$, and (d) $r = 4$. One can also notice two temporal windows, at least in (a), (b), and (c): a transient part during which the trajectory evolves from an initial condition, and an asymptotic part where such a transient is not visible anymore, since the system has reached some asymptotic stable state (steady, periodic, or chaotic).

1.1.3 Bifurcation diagram

Qualitative changes in dynamical systems which are the consequence of the variation of parameters on which they depend on, are defined as bifurcations. Then, in our population growth example Eq. (1.2), the graphical representation of all solutions x_n as functions of r is known as bifurcation diagram. Figure 1.1 shows the bifurcation diagram of the logistic map, where one can find fixed point, periodic and chaotic solutions of its difference equation. Each solution is described in the following:

Stationary solutions consist in time series that do not evolve in time. These trajectories converge asymptotically to a fixed point after an initial transient generated by the introduction of initial conditions in dissipative systems, see Fig. 1.2(a). These kind of solutions maintain such state until they are perturbed by an external force, or they are modified.

Periodic solutions correspond to sequences which converge to a set of

values that repeats cyclically, see Fig. 1.2(b). As an important remark, if solutions come from a dissipative dynamical model, they are known as *limit cycles*. In such models, the non-linearity appears in the dissipation coefficient of the equation, which is essential to maintain the amplitude of the periodic solution [13]. Additionally, we can also find periodic solutions with twice the period of the previous solutions, see Fig. 1.2(c). These kind of bifurcations can be found in discrete and continuous-time dynamical systems, and they are known as period doubling bifurcations [9, 10].

Chaotic solutions are aperiodic and a-priori unpredictable solutions, where the unpredictability is related with the impossibility to determine the system's development for all future. An essential property of chaotic systems is the sensitivity to initial conditions, that is related to the exponentially divergent evolution of two trajectories infinitesimally close to each other at the beginning. Therefore, if the system is chaotic, any arbitrarily small separation between initial conditions is going to exponentially increase in time. In Fig. 1.2(d), two chaotic solutions of the same logistic map are shown for two infinitesimally different initial conditions. As it can be seen, initially both signals maintain the same evolution, but then at some point they start diverging exponentially.

Despite the fact that chaotic solutions appear in 1-dimensional discrete-time dynamical system, such as the logistic map, additional conditions are required for continuous-time systems. In the last case, a minimum of 3 dimensions are required in a continuous-time nonlinear system to generate chaotic solutions.

1.2 Continuous-time chaotic systems

1.2.1 Lorenz and Rössler chaotic attractors

The Lorenz system (1963) generated the first numerically observed continuous-time chaotic solutions observed in the history of dynamical systems theory [14, 11]. It emerged from the field of atmospheric sciences, as a 3-dimensional model of climate fluctuations. The system of equations is the following,

$$\begin{aligned} \dot{x} &= a_1(y - x), \\ \dot{y} &= b_1x - y - xz, \\ \dot{z} &= xy - c_1z; \end{aligned} \tag{1.5}$$

where a_1 is the Prandtl number related with the fluid viscosity, b_1 is the Rayleigh number related with the difference of temperature between two surfaces, and c_1 is a scaling factor. The nonlinearity comes from the multiplication

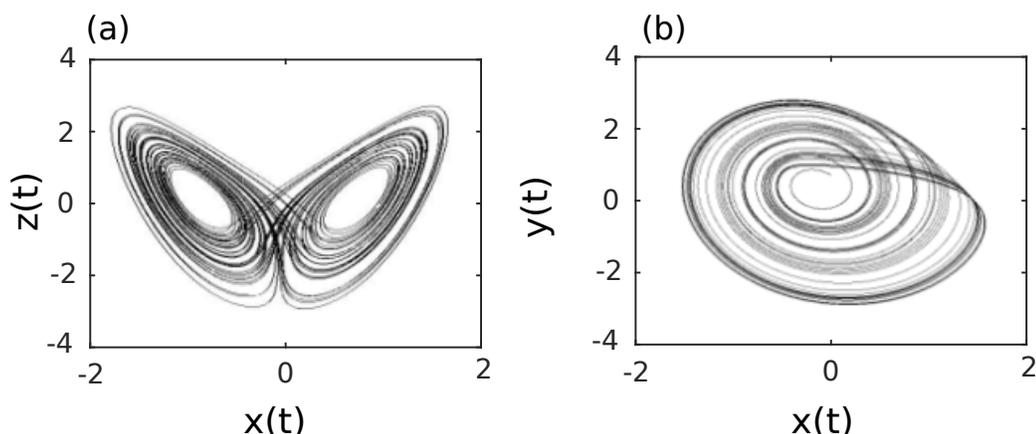


Figure 1.3: (a) Lorenz and (b) Rössler chaotic attractors.

between two different variables $\{xz, xy\}$. In fact, the solutions to this nonlinear system were possible to be numerically obtained at a time when the available computational power was just enough to numerically solve a system of 3 nonlinear differential equations. This was achieved by increasing the number of bits from 3 to 6 in the numerical simulation, consequently the accuracy of calculations was increased, and the chaotic solutions appeared. In general, as continuous-time chaotic oscillators have no analytical solutions, a numerical approximation is always required to obtain an approximate solution to their system of equations.

Figure 1.3(a) shows the solutions of two dynamical variables $\{z, x\}$ from the Lorenz system, solved by using a Runge-Kutta method of 4th order. The space where these variables are represented is defined as the phase space. The **phase space** (also state space) is the geometric space where the evolution of the system's state trajectory is represented, i.e. with all the necessary dimensions or degree of freedom of the original dynamical system. Each state is constructed by the contribution of all N independent variables $\{x^{(i)}\}_N$, which describe the dynamics of a N -dimensional system. The trajectory (or orbit) in phase space is in reality a sequence of states $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, that shows the dynamics of the system until time step n , with $\mathbf{x}_j = (x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(N)})$. In the current example, the independent variables are $x^{(1)} = x; x^{(2)} = y; x^{(3)} = z$, then the phase space is 3-dimensional, and consequently each state of the Lorenz attractor is defined by 3 components.

The double-scroll object shown by Fig. 1.3(a) is a 2-dimensional projection of

a 3-dimensional geometrical object, defined as an attractor. **Chaotic attractors** are trajectories formed by states which are being attracted to a specific region of the phase space where they remain confined to. In other cases, trajectories can also escape from one region of the phase space, indicating the presence of a *repellor* [9].

Thirteen years after the discovery of the double-scroll chaotic attractor by E. Lorenz, Otto Rössler showed that a set of simple mathematical relations was capable to generate another trajectory also exhibiting a chaotic signature. This system's purpose was to model a chemical process, which consists in the coupling between a linear chemical oscillator and a system exhibiting chemical hysteresis,

$$\begin{aligned}\dot{x} &= -(y + z), \\ \dot{y} &= x + a_2y, \\ \dot{z} &= b_2 + xz - c_2z;\end{aligned}\tag{1.6}$$

the variables describing concentration of chemical substances x and y come from the linear oscillator and z from the hysteretic one. The constants a_2 , b_2 , c_2 are effective reaction rates [15, 16]. This new attractor found by Rössler contains just a nonlinear component made from the multiplication of two variables xz , and it consists in one single-scroll, see Fig. 1.3(b).

Until this point, we have introduced Lorenz and Rössler oscillators as representative examples of three dimensional continuous-time chaotic attractors. However, there are other ways to obtain continuous-time chaotic solutions, coming from models consisting of one nonlinear ordinary differential equation (ODE). One of those ways includes the introduction of an explicit temporal variable to the ODE, such as a time delay with respect to the main temporal variable t . We define these models as delay differential equations (DDEs), where a DDE is in fact equivalent to an infinite-dimensional system of differential equations [17]. The number of solutions to a DDE are in theory infinite due to the infinite amount of initial conditions in the continuous rank required to solve the equation. Each initial condition initializes an ODE from the infinite-dimensional system of equations. Consequently, the introduction of a time delay in an ODE, resulting in a DDE, thus provides sufficient dimensionality to allow for the existence of chaotic solutions.

1.2.2 Mackey-Glass system

As an historical remark, DDEs were considered as simple nonlinear mathematical models to describe a large class of human diseases and pathologies. For example, DDEs reproduce the phenomena generated by an abnormal respi-

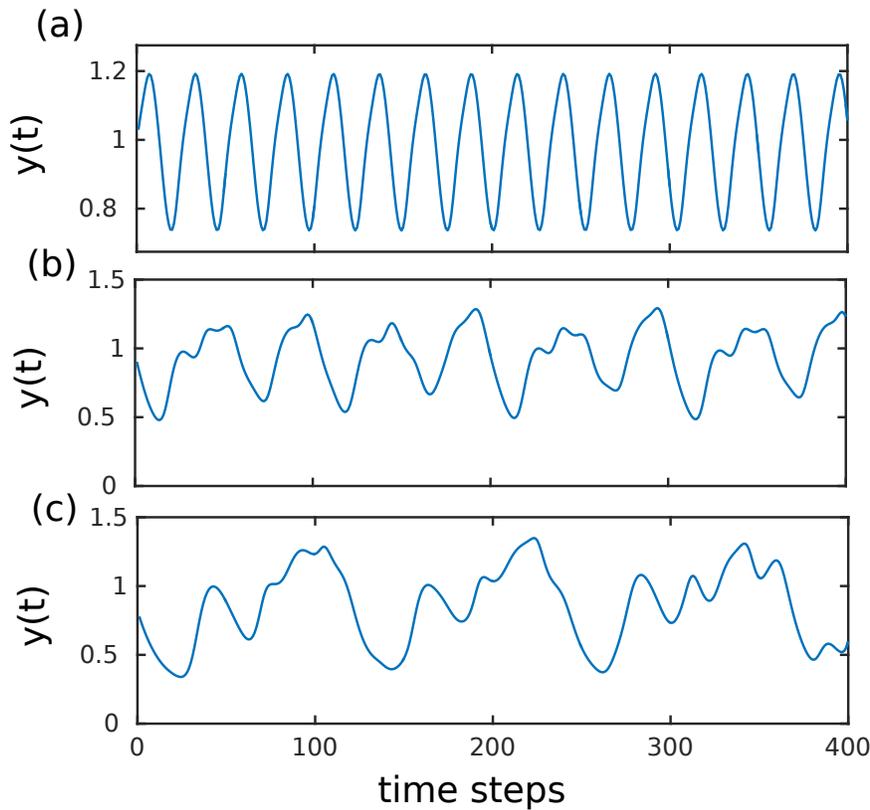


Figure 1.4: Mackey-Glass chaotic time series (set around 0 by subtracting 1 to the original amplitude) for time delay τ_m : (a) 8, (b) 17, and (c) 34.

ration process, where the increase of the time delay is associated with high amplitude oscillations leading to apnea (temporary cessation of breathing). Such abnormal respiration process can turn chaotic, corresponding to a pathological condition in humans. Another example of DDEs used in physiology is the Mackey-Glass (MG) system, which attempts to model the density of a homogeneous population of mature circulating cells. The delay term is associated to the time-gap between the initiation of blood cell formation in the bone marrow and the release of mature cells.

The MG system is a first order nonlinear DDE, described as follows [18],

$$\frac{dy}{dt}(t) = \vartheta \frac{y(t - \tau_m)}{1 + [y(t - \tau_m)]^\nu} - \psi y(t), \quad (1.7)$$

where parameters ϑ, ν, ψ are real numbers, and τ_m is the time delay. The

time-discrete version of the MG system is the following [111]:

$$y_{n+1} = y_n + \delta \left(\frac{\vartheta y_{\tau_m}}{1 + (y_{\tau_m})^\nu} - \psi y_n \right), \quad (1.8)$$

where $y_{\tau_m} = y(n - \tau_m/\delta)$, $\tau_m = 17$, and $\delta = 1/10$ is the stepsize indicating that the time series is subsampled by 10. Where one increment of the resultant time series correspond to 10 time units of the Mackey-Glass equation. The other parameters are typically set to $\vartheta = 0.2, \nu = 10, \psi = 0.1$. If τ_m is increased the system bifurcates, allowing the existence of stable periodic solutions and chaotic behavior. In Fig. 1.4(a,b,c), the increase of the time delay $\tau_m = \{8, 17, 34\}$ in Eq. (1.7) results in dynamics evolving from periodic to chaotic behavior. Chaotic responses are related to high values of time delay ($\tau_m > 15$), which can be interpreted as indicators of that the structure under study is out of control. In our physiological example, this large delay could lead to problems in the release of mature cells on time, relevant for the case when they would have to replace defect cells.

Under such circumstances, chaotic solutions seem to describe complex and irregular behavior in natural oscillators. But, what about some other kind of oscillators designed by humans? Therefore, we introduce an additional 1-dimensional system with time delay, belonging to the optics field. Such a model, introduced in the following section, is capable of generating dynamics at least as complex as the MG system, i.e. chaotic solutions.

1.2.3 Ikeda system

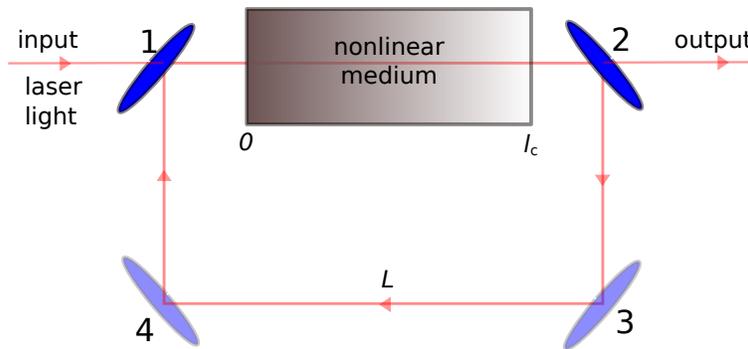


Figure 1.5: Scheme of a ring resonator model.

The Ikeda system describes the dynamics of a ring cavity resonator containing a nonlinear dielectric medium. A continuous-wave beam, coming from a

laser located at the input, enters at the partially transmitting mirror 1 (with reflectivity R) and propagates through a nonlinear dielectric medium of length l_c . This medium has a nonlinear refractive index η and absorption coefficient ρ . The lightpulses inside this medium have wave number k_w . After the dielectric medium, the light propagates to mirror 2 (with reflectivity R), then it completes a lossless round-trip in the cavity of length L through reflections on mirrors 3 and 4 (with 100% reflectivity), see Fig. 1.5.

The optical field at the output is the following [19, 20],

$$E(t) = A + BE(t - t_R)e^{j[|E(t-t_R)|^2 - \phi_0]}, \quad (1.9)$$

where $A = \{(1 - R)k_{wl}|\eta_2|(1 - e^{-\rho l_c})/\rho\}^{1/2}E_I$ is a parameter proportional to the incident field amplitude, η_2 is the quadratic coefficient of the nonlinear refractive index, ϕ_0 constant offset, and E_I is the amplitude of the incident field. $B = Re^{-\rho l_c} (< 1)$ is a parameter that describes the dissipation in the cavity, with c as the speed of the light in vacuum. The time delay $t_R = l_c/c$ is originated from the finite propagation speed of the light in the cavity. Equation (1.9) can be written as $E(t) = \Omega(E(t - t_R))$, whose time evolution approximately follows the mapping rule: $E_{n+1} = \Omega(E_n)$. Then, after many iterations of E_{n+1} , it is possible to show that the intensity $I_n = |E_n|^2$ obeys the following 1-dimensional map:

$$I_{n+1} = A^2[1 + 2B \cos(I_n - \phi_0)] \equiv f(A^2; I_n). \quad (1.10)$$

We furthermore include information about relaxation time of the nonlinear medium:

$$\phi(t) = \gamma \int_{-\infty}^t e^{-\gamma(t-s)} |E(s-t_R)|^2 ds, \quad (1.11)$$

where γ is the relaxation rate. Afterwards, we isolate the nonlinear phase shift $|E(t - t_R)|^2$, and replace it in Eq. (1.9). For the limit where the dissipation in the cavity is small, $B \ll 1$, we obtain a DDE describing the time evolution of the relaxation process,

$$\gamma^{-1} \dot{\phi}(t) = -\phi(t) + f(A^2; \phi(t - t_R)). \quad (1.12)$$

Therefore, combining Eqs. (1.10) and (1.12) we obtain the classical Ikeda DDE [19],

$$\gamma^{-1} \dot{\phi}(t) = -\phi(t) + A^2[1 + 2B \cos(\phi(t - t_R) - \phi_0)]. \quad (1.13)$$

Due to the presence of the time delay and the condition $t_R \gg \gamma^{-1}$, the Ikeda system is capable to generate chaotic solutions and multiple stabilities [19, 20].

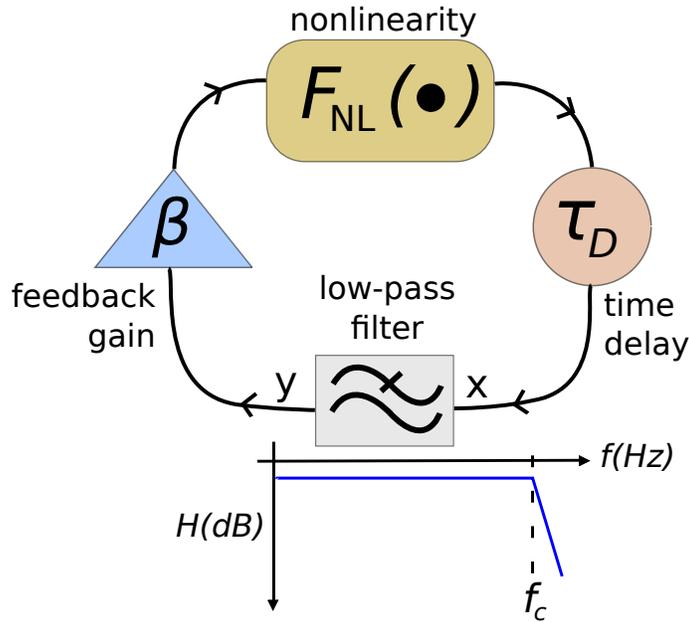


Figure 1.6: Block diagram of a ring resonator model composed by a nonlinear function, time delay, low-pass filter and feedback gain.

From a different point of view, a similar mathematical equation is found by combining a nonlinear function, time delay and a filtering block in a feedback close loop, shown by Fig. 1.6. This scheme attempts to generalize the generation of chaotic solutions in a simple structure constituted by 4 blocks. At first, we consider a set of values $y(t)$ passing through a nonlinear function $F_{NL}(\cdot)$, time delayed by τ_D and amplified by β . The relation which links $y(t)$ to the output of the blocks of feedback gain, nonlinearity, and time delay is:

$$x(t) = \beta F_{NL}[y(t - \tau_D)]. \quad (1.14)$$

The outgoing information is filtered by a low-pass filter, which attenuates high frequencies and lets low frequencies unattenuated. The response time of the filter is given by $\tau = 1/2\pi f_c$, with f_c as the cutoff frequency. The parameter τ indicates the minimum time that the Ikeda system spends in a transition after a perturbation is applied.

The effect of the filter can be defined as a convolution between the impulse response $h(t)$ and the input $x(t)$. The output of the filter is therefore:

$$y(t) = x(t) * h(t), \quad (1.15)$$

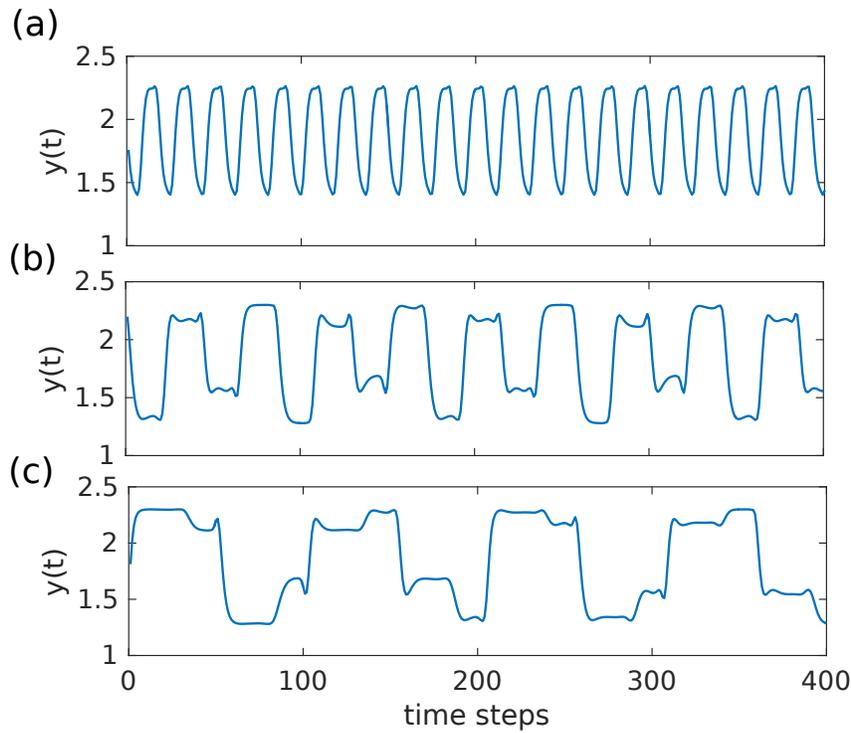


Figure 1.7: Ikeda chaotic time series for time delay τ_D : (a) 8, (b) 20, and (c) 50.

where

$$y(t) = \int_{-\infty}^{\infty} x(t - \theta)h(\theta)d\theta. \quad (1.16)$$

In the Fourier space, we represent the temporal variables $\{x(t), y(t), h(t)\}$ as frequency variables $\{X(\omega), Y(\omega), H(\omega)\}$, with transfer function $H(\omega)$ of the low-pass filter written as follows:

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{1}{1 + j\omega\tau}. \quad (1.17)$$

The property of the Fourier transform:

$$j\omega Y(\omega) = \text{FT} \left[\frac{dy(t)}{dt} \right], \quad (1.18)$$

is used in Eq. (1.17) to obtain the equation of the filter in the frequency domain,

$$Y(\omega) + \text{FT} \left[\frac{dy(t)}{dt} \right] \tau = X(\omega). \quad (1.19)$$

which is represented in the temporal domain as a first order differential equation:

$$y(t) + \tau \frac{dy(t)}{dt} = x(t). \quad (1.20)$$

As shown in Fig. 1.6, the output $y(t)$ of the filter is connected to the blocks where the feedback gain, nonlinearity and time-delay τ_D are applied. Thus, if $F_{NL}(\cdot) = \cos(\cdot + \phi_0)$, we substitute Eq. (1.14) in Eq. (1.20), and we obtain,

$$y(t) + \tau \frac{dy(t)}{dt} = \beta \cos[y(t - \tau_D) + \phi_0]. \quad (1.21)$$

This equation is a modified version of the classical Ikeda system, Eq. (1.13). Some of the dynamics generated by this equation for the parameters $\tau = 0.7958$, feedback gain $\beta = 2.3$ and offset $\phi_0 = -\pi/2$ can be seen in Fig. 1.7. Here, the time delay τ_D takes the values $\{8, 20, 50\}$, and the corresponding solutions of the Ikeda equation are shown by Fig. 1.7(a,b,c), respectively. The solutions for each time delay change from periodic ($\tau_D = 8$) to chaotic ones ($\tau_D = \{20, 50\}$).

1.3 Complexity measure and time series analysis

As previously mentioned in Sec. 1.2.1, since chaotic systems have no analytic solutions, we need to search for methods to approximate their solutions. In all previous cases, we utilized numerical methods to solve systems of nonlinear equations. However, there are many cases where we do not have access to the equations which model the oscillator under study, i.e. the source of these solutions is unavailable. That is to say, when the only information that we have about the oscillators is their solutions, we have to search for methods with which to obtain additional information about their sources. In this section, we introduce a set of qualitative and quantitative methods to analyse the dimensionality and complexity of time series coming from nonlinear hidden sources.

1.3.1 Attractor reconstruction

In practice, it is not always feasible to collect information of all state space dimensions. For example, in the MG and Ikeda oscillators, one just has access

to a time series from a single accessible variable, while all others remain hidden. Nevertheless, hidden variables are participating in the development of the global dynamics as well as the accessible variables. In order to approximate a full state-space representation of the oscillators, we have to embed the 1-dimensional sequence into a high dimensional space. The condition to embed in such space is at first to identify the number of dimensions of the original oscillator, with which we ensure that the embedded time series has the same dimensionality of the source. One of the most practised methods to embed 1-dimensional information is given by the Takens embedding theorem [21].

Takens embedding theorem

Theorem 1. *The time delayed version of one time series suffices to reveal the structure of an attractor. Let us represent the data in M -dimensional space by the vector $\mathbf{y} = [y_n, y_{n-\tau_0}, \dots, y_{n-(M-1)\tau_0}]^\dagger$. Where $(\cdot)^\dagger$ stands for matrix transposition. The pair dimension-delay for the embedding (M, τ_0) contributes to reconstruct the right object in the state space.*

The Takens scheme introduces time delay τ_0 to the original sequence in order to create delay coordinates, that span the dimensionality M of the original oscillator. The time delay is estimated with an *autocorrelation analysis* (AC) to the input, where τ_0 is the first minimum in the AC function [22],

$$R_{yy}(l) = \lim_{g \rightarrow \infty} \frac{1}{g} \sum_{n=0}^{g-1} y_n y_{n-l}^*, \quad (1.22)$$

with l as the time lag, and g the length of the dataset. The estimation of the first minimum in the AC function maximizes the possibility to obtain linearly independent observations from the system, which are related with the inference of the degrees of freedom behavior of the chaotic original oscillator [21, 23, 24]. In order to obtain full-spatial representations of the MG and Ikeda attractors, we reconstruct the state space according to Takens scheme by finding at first the value of τ_0 . Figure 1.8(a,b) shows that the absolute value of the autocorrelation functions for a set of 10000 values from the MG ($\tau_m = 17$ in Eq. (1.7)) and 6000 values from the Ikeda ($\tau_D = 20$ in Eq. (1.20)) systems have their first minima at lags $l = \pm 12$ and $l = \pm 11$, respectively.

After having identified the embedding delay for each model, the question about how many embedding dimensions are required to fully reconstruct the attractor still remains open. The minimum amount of coordinates that are required to embed the system under study can be estimated by the *false nearest*

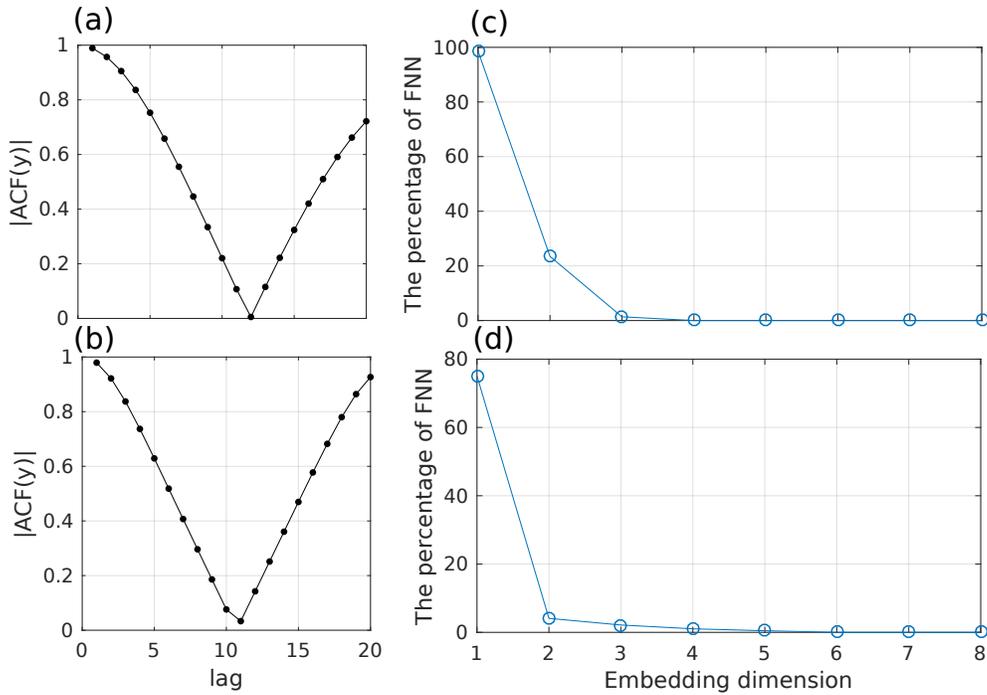


Figure 1.8: Absolute value of the autocorrelation functions for (a) MG and (b) Ikeda. False nearest neighbors analysis for (c) MG and (d) Ikeda.

neighbors (FNN) method. This method consists in reconstructing the attractor's state space using different dimensionalities, e.g. M_1 and $(M_1 + 1)$ [25]. If the embedding dimensions, M_1 and $(M_1 + 1)$, are not sufficient to describe the full dynamical system's dimensionality, the projection of any state present in the M_1 -space is not going to fall in the neighborhood of its twin in the $(M_1 + 1)$ -space. Then, the state with dimension M_1 is a false neighbor of its pair in the $(M_1 + 1)$ -state space. Consequently, it is required to discretely change M_1 until we detect the minimal dimension of the system, where there are no more false neighbors. One popular method to estimate the dimensionality is by the reconstruction of two state-spaces with different dimensions as it was described above. One then creates grids which define the neighborhood of each point in the state-space, and compare them according to

$$R_{\mathbf{y}_n} = \frac{\|\mathbf{y}_{n+1} - \mathbf{y}'_{n+1}\|}{\|\mathbf{y}_n - \mathbf{y}'_n\|}, \quad (1.23)$$

where states \mathbf{y}_n and \mathbf{y}'_n , are in M_1 and $(M_1 + 1)$ -spaces, respectively.

In practice, we increase M_1 from one up to the value where no more false

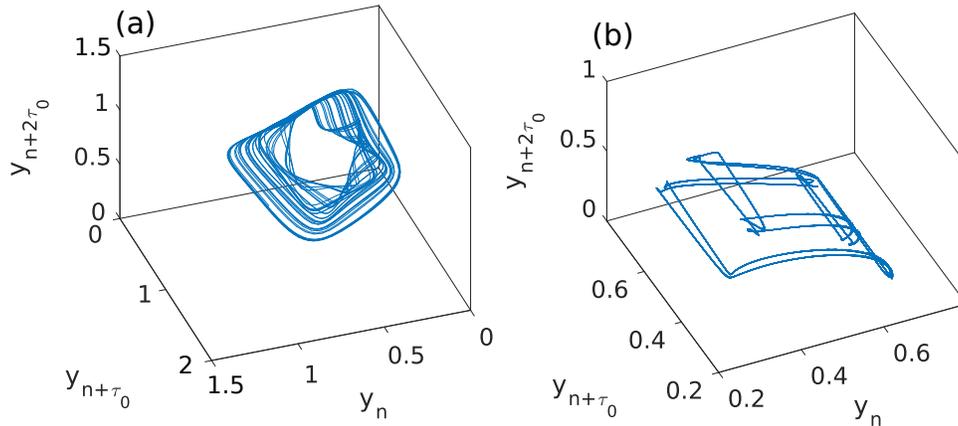


Figure 1.9: Three dimensional Takens embedding of (a) MG for $\tau_m = 17$ and (b) Ikeda for $\tau_D = 20$ attractors.

neighbors are found. Then, for each pair of (M, τ_0) , the data y_n is embedded and we determine the nearest neighbor to each state. If the ratio of next $\|\mathbf{y}_{n+1} - \mathbf{y}'_{n+1}\|$ and current $\|\mathbf{y}_n - \mathbf{y}'_n\|$ states is greater than $R_{\mathbf{y}_n}$, then they are false neighbors. In the simulation we set $R_{\mathbf{y}_n} = 5$, and then we estimated the percentage of FNN found per each dimension. The results are shown by Fig. 1.8(c,d) for 10000 values from the MG ($\tau_m = 17$ in Eq. (1.7)) and 6000 values from the Ikeda ($\tau_D = 20$ in Eq. (1.20)) systems. The minimal dimensions of these systems are $M = 4$ and $M = 7$, respectively. Therefore, we can reconstruct the attractors by building coordinates with lags $l = \{\pm 36, \pm 24, \pm 12, 0\}$ for MG system, and $l = \{\pm 66, \pm 55, \pm 44, \pm 33, \pm 22, \pm 11, 0\}$ for Ikeda system. In Fig. 1.9 we show the 3D projections of the embedded MG and Ikeda attractors in panel (a) and (b), respectively.

The here presented chaotic attractors are defined in limited regions of their state spaces, forming well-defined geometrical objects with distinctive shapes. Nevertheless, if we choose two points from neighbor trajectories in the attractors, it can be demonstrated that this set of points are going to take very different paths after some time. Such divergence between points in neighbor trajectories is quantified through the Lyapunov exponent.

1.3.2 Lyapunov exponent

The Lyapunov exponent λ of a dynamical system quantifies the averaged rate of exponential divergence (or convergence) between neighbor trajectories. If

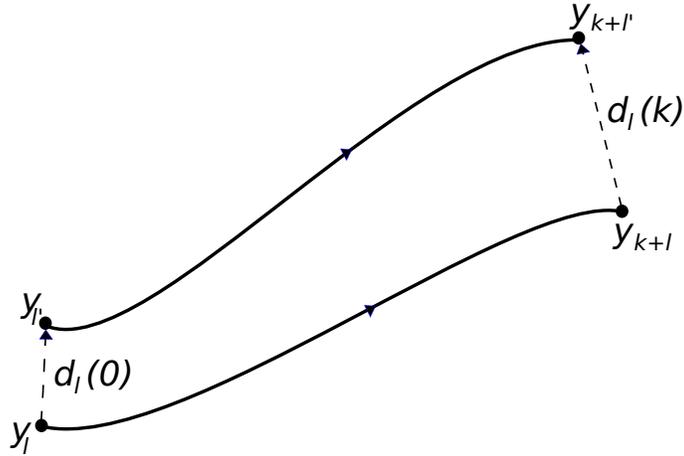


Figure 1.10: Illustrative example of the divergent evolution of two trajectories from the same chaotic system in phase space, that were initially infinitesimally closer to each other.

the system under study is chaotic, then the initial distance $d_l(0)$ between two trajectories $\{\mathbf{y}_l, \mathbf{y}_{l'}\}$ at time $n = 0$ will have an exponentially divergent evolution with final distance $d_l(k)$ between the two trajectories $\{\mathbf{y}_{l+k}, \mathbf{y}_{l'+k}\}$ at time $n = k$, see Fig. 1.10. In fact, there are as many Lyapunov exponents as dimensions have the system under study. In a chaotic system at time $n = 0$, if we define our initial distance $d_l(0)$ in a sphere where we can find all initial states, then at time $n = k$ that distance will belong to an ellipsoid, and it will be defined by $d_l(k)$. So the diameter of this ellipsoid is defined by the most positive Lyapunov exponent, which is the largest λ .

The value of the largest Lyapunov exponent λ_{max} is estimated by the following approach [26, 27]: we reconstruct the state space via Takens embedding theorem [21], where a reconstructed trajectory will be represented by a set of S states in total. The next step consists in find the nearest neighbor $\mathbf{y}_{l'}$ for each point \mathbf{y}_l on the trajectory. It can be done by searching for the point that minimizes the distance $\|\mathbf{y}_l - \mathbf{y}_{l'}\|$ at time $n = 0$. The largest Lyapunov exponent estimates the distances between these two points at time $n = k$, according to [27]

$$\lambda_{max}(k) = \frac{1}{k\Delta n(S-k)} \sum_{l=1}^{S-k} \frac{d_l(k)}{d_l(0)}. \quad (1.24)$$

Here $d_l(0) = \min_{\mathbf{y}_{l'}} \|\mathbf{y}_l - \mathbf{y}_{l'}\|$ is the initial distance between point \mathbf{y}_l to its neighbor $\mathbf{y}_{l'}$, $d_l(k) = \min_{\mathbf{y}_{l'+k}} \|\mathbf{y}_{l+k} - \mathbf{y}_{l'+k}\|$ is the distance from the point \mathbf{y}_{l+k}

to its neighbor point $\mathbf{y}_{l'+k}$ after k discrete time steps n . From this analysis, we can characterize dynamical evolutions by λ_{max} . In table I.I, we give a list of the dynamics that we can relate to λ_{max} values, starting from fixed point solutions until noise-like (hyper-chaotic) states.

Table I.I: Largest Lyapunov exponent

Dynamics	largest Lyapunov exponent
Fixed point	$\lambda_{max} < 0$
Limit cycle	$\lambda_{max} = 0$
Chaos	$0 < \lambda_{max} < \infty$
Noise	$\lambda_{max} \rightarrow \infty$

For the MG and Ikeda systems, we estimate λ_{max} for a set of 10000 values from the MG ($\tau_m = 17$ in Eq. (1.7)) and 6000 values from the Ikeda ($\tau_D = 20$ in Eq. (1.20)) systems with ($M = 4, \tau_0 = \pm 12$) and ($M = 7, \tau_0 = \pm 11$), respectively. The MG system has $\lambda_{max} = 0.0036$ and Ikeda $\lambda_{max} = 0.0465$.

1.3.3 Mutual information

Mutual information (MI) is a measure of dependency between time series generated by nonlinear dynamical systems. It quantifies the amount of information shared by two random variables x, y [28]. It is defined as

$$MI = \sum_{x,y} \Psi(x,y) \log \frac{\Psi(x,y)}{\Psi(x)\Psi(y)}, \quad (1.25)$$

where $\Psi(x,y)$ is the joint probability density function of x and y , while $\Psi(x), \Psi(y)$ are probability density functions of x and y , respectively. In the case where $MI = 0$, the random variables x and y are independent, no information is shared between x and y . If $MI \gg 0$, the random variables x and y share information between each other, and it is possible to extract information corresponding to variable x from variable y . In practice, the probability density functions Ψ are estimated with histograms.

1.4 Neural networks

Up to now, we were studying autonomous dynamical systems without external variables. Typically, autonomous systems evolve according to their own prescribed causal rule. On the other hand, non-autonomous (or driven) systems

are coupled to any external variables perturbing their spontaneous activity. In this work, we introduce the framework of driven systems, which includes multiple classes of complex structures such as neural networks.

In the early 1940s, the first artificial neural network model appears as an attempt to model the processes in the brain [29]. Those attempts are mainly related to the brain capabilities to solve multiple complex tasks and general purpose information processing. In the cases where such models of the brain are based on recurrent neural networks, they can be studied under the principles of dynamical systems theory [2]. Then, we could think the brain as an autonomous dynamical system that is continuously being perturbed by information to be processed by it. Consequently, the brain structure in this case can be interpreted as a part of the driven systems framework.

Before we go deep into neural networks within the driven systems framework, let us introduce a summary of how biological and artificial neural networks are designed in general.

1.4.1 Biological and artificial Neurons

A biological neuron is a cell composed by dendrites, body, axon and synaptic terminals, see the schematic illustration in Fig. 1.11(a). The dendrites carry input signals into the cell body, where this incoming information is summed to produce a single reaction. In most cases, the transmission of signals between neurons are represented by action potentials at the axon of the cell, which are changes of polarization potential of the cell membrane. The anatomic structure where the neurons communicate with each other is known as synapse [30]. The cell membrane has a polarization potential of -70 mV at resting state, produced by an imbalanced concentration inside and outside of its charged molecules. This change of the polarization happens when several pulses arrive almost simultaneously at the cell. Then, the potential increases from -70 mV to approximately $+40$ mV. Some time after the perturbation, the membrane potential becomes negative again but it falls to ≈ -80 mV [2, 3]. The cell recovers gradually, and at some point the cell membrane returns to its initial potential (-70 mV), as schematically illustrated in Fig. 1.11(b).

An artificial neuron is shown in its general model by Fig. 1.11(c). The output y of the neuron represents the signals coming from the axon, and it is mathematically described by

$$y = f(\mathbf{W} \cdot \mathbf{x} + W^{off}b). \quad (1.26)$$

The x^i inputs transmit the information to the neuron through the weights

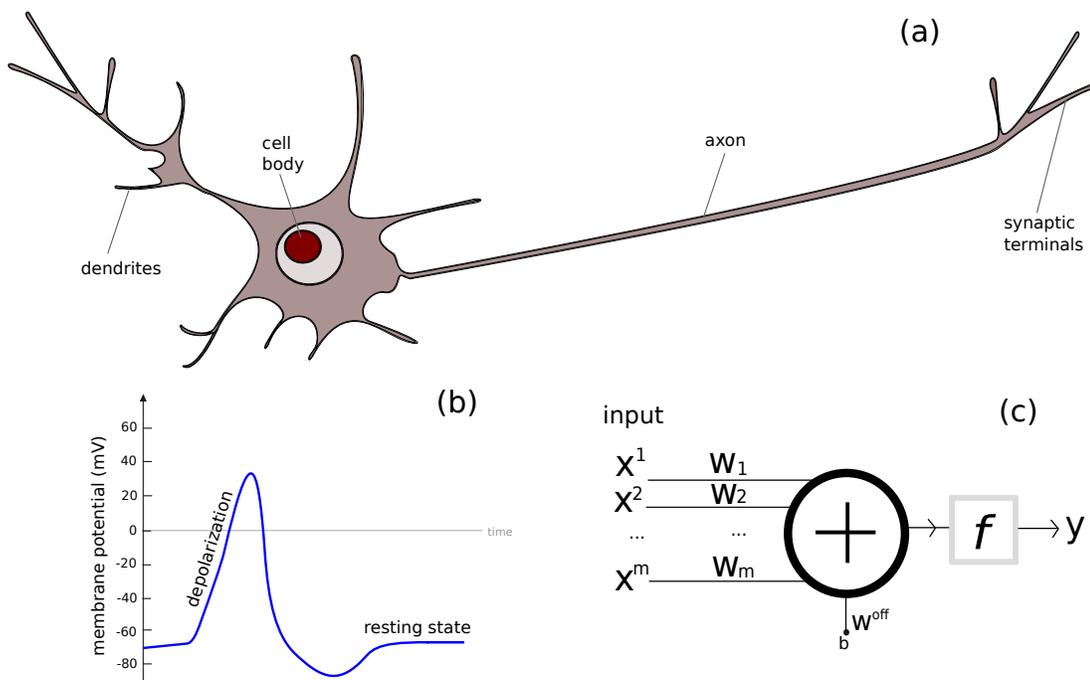


Figure 1.11: (a) Diagram of a biological neuron. (b) Plot of a typical action potential of a neuron. (c) Scheme of the mathematical model of a neuron.

W_i , which correspond to the strength of the synapses. The summation of all weighted inputs, and their transformation via activation function f , are associated with the physiological role of the cell body. The activation function can be linear or nonlinear. The bias b represents an extra variable that remains in the system even if the rest of the inputs are absent. The weight of the bias, W^{off} , is usually constant. The bias can also be seen as an offset for varying the operating point of the activation function.

1.4.2 Artificial neural networks

Among many other categorization, artificial neural networks (ANNs) can be categorized in the following two main branches:

Feedforward neural networks

A simple model of an ANN is presented in Fig. 1.12(a) [3]. It is constituted by a network of m neurons with p inputs. The information is distributed along

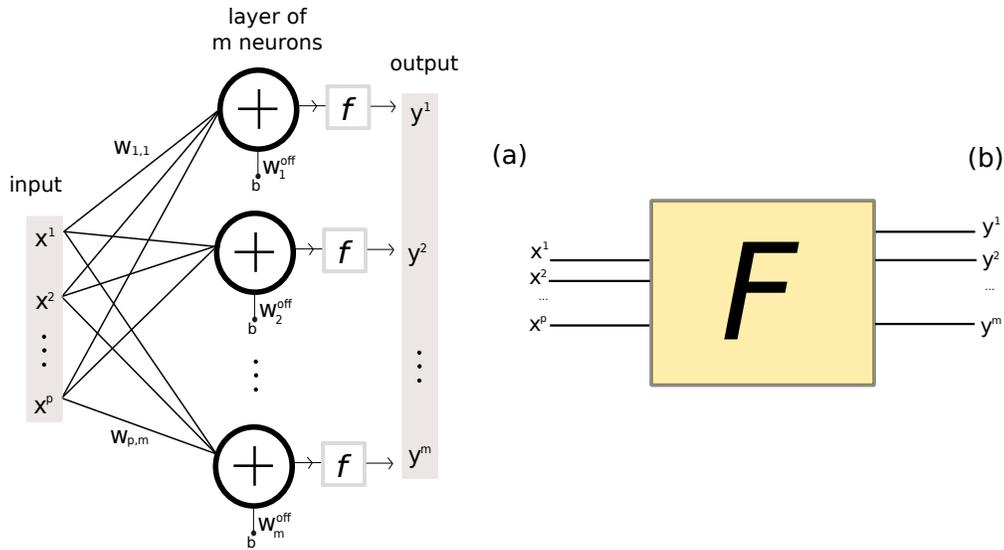


Figure 1.12: (a) Scheme of a single layer of m neuron models, where each neuron has p different input links. (b) Illustrative block diagram of the blackbox interpretation of ANNs.

the neurons through internal weights $W_{i,j}$. As introduced, the role of ANNs is to mimic the information processing capabilities of the brain. However, it still remains unknown how the brain works, causing it to be seen as a blackbox that solve complex problems. That is to say, we provide information to the brain through its inputs and we get a desired output [31]. Feedforward ANNs imitate these capabilities by building a model F , performing smooth function fits to input information. In this sense, the network's functionality can be interpreted as a static function $F : \mathbb{R}^p \rightarrow \mathbb{R}^m$, mapping the information $x_i \in \mathbb{R}^p$ into $y_j \in \mathbb{R}^m$, see Fig. 1.12(b). This process can also be thought as a kind of linear/nonlinear regression [2].

In many cases, feedforward ANNs are designed with more than one layer of neurons, see Fig. 1.13(a). Here, we present a model with s layers which attempts to solve highly complex problems previously deemed unsolvable, such as pattern recognition, classification, and prediction [32, 33, 34]. The use of several layers allows the network to have a rich ensemble of representations of data internal features [35]. Despite the fact that feedforward ANNs are powerful tools for information processing, they usually take long time to find solutions of complex tasks. For example, its optimization process is typically performed by modifying the ANN's topology [36], where often optimizing

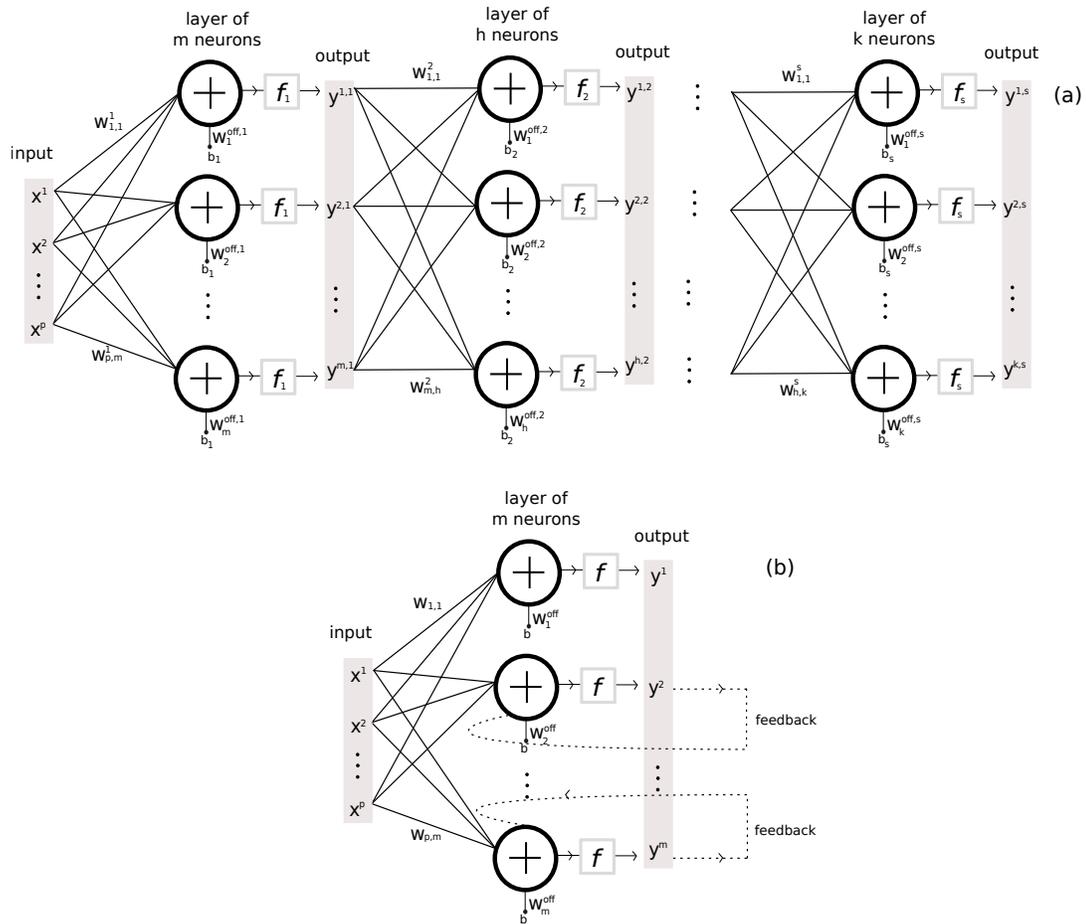


Figure 1.13: Scheme of: (a) several layers of neurons connected to each other via input-output links; (b) recurrent neural network.

global performances is time consuming [37, 38].

Feedforward ANNs have a unidirectional flow of information inside the network, without any feedback. They behave like finite impulse response (FIR) digital filters, which compensate this “finiteness” usually with a very high numbers of layers and neurons. This architecture is known as “deep learning”, which has gained popularity and recognition due to its amazing performances [39]. However, at the cost of extreme computational power (in memory size) and processor speed.

Recurrent neural networks

On the other hand, recurrent neural networks (RNNs) are representing a conceptually very different configuration, involving internal feedback in the network, and thus complex internal dynamics even with a low number of neurons [35]. To repeat the analogy with digital filters, one would have to deal with infinite impulse response filters (IIR), which are not unconditionally stable, unlike FIR. Figure 1.13(b) shows a scheme of a RNN, that has the same configuration of a feedforward ANN, but with feedback connections. Feedback elements allow the information to be recycled, transformed and reused. This implies memorization with which it is possible to generate internal dynamics [2]. They therefore have an explicit temporal variable on addition to the spatial variables in their mathematical models.

In fact, if the activation functions are nonlinear, networks have to be seen as nonlinear dynamical systems. Consequently, the stability of the system might play a fundamental role for information processing. The input is mapped into the high dimensional state-space of the dynamical system, then the solutions are associated to a particular state of it. The information is usually injected when there is a stable equilibrium point, or even near criticality [40, 41].

The typical applications of RNNs are: associative memories [42] and time series prediction [5]. The associative memories are used for recognition of trained input patterns, where an exemplary noisy pattern can be associated to a stored pattern in the network. In time series prediction, it is possible to use RNNs to mimic the a-priori unpredictable behavior of some feedback systems, such as chaotic oscillators. One of the main limitations of these models is that some RNNs require convergence to a stable state after the injection of the input signal, i.e. they require a process of convergence at some point. If this stable state is not reached, the network can store highly distorted patterns, then it could not solve the tasks based on what it has memorized.

1.4.3 Training

The design of a network for a particular task is accompanied by the adaptation of the network's elements. Weights and biases have to be adjusted in order to solve a specific task. This procedure is known as training (or learning) the network. The adaptation of the network's parameters to the problem usually follows a learning rule. In this context, this means that weights and biases are changed in a controlled manner according to some learning rule. One can identify three kind of learning methods: supervised, reinforcement and unsupervised [35].

- **Supervised:** The network is trained by using a teacher/target, which is used to adjust the weights to generate an output similar to the target. This learning rule minimizes the difference between teacher and output. The training performance is measured by comparing the teacher with the output, and the correct outputs referred to as the teacher.

- **Reinforcement:** This learning rule is similar to the supervised rule, with the exception that there is no a restriction to find only one correct output. Here, the performance is measured in a more qualitative manner, by assigning scores (or grades) which provide answers about how good or bad is the approximation of the network's output with respect to the teacher.

- **Unsupervised:** In this case there is no any teacher/target externally provided. Weights of the network are modified via self-organization as a response to the perturbation with the input information. The self-organization feature corresponds to the network's propensity to cluster input patterns. This clustering-like training allows the categorization of inputs of the same nature into a specific unique cluster.

Training methods

Some of the most common learning rules are listed in the following:

- **Gradient descent:** As it was described previously, the weights of a network should be adapted to solve a particular task. Consequently, the error function of the network Ξ , which characterize the network's task performance, has to be minimized. The error function is a function of the weights, i.e. $\Xi = \Xi(W_1, W_2, \dots, W_p)$. One way to minimize Ξ is at first by estimating the gradient of Ξ [2],

$$\nabla\Xi = \left(\frac{\partial\Xi}{\partial W_1}, \frac{\partial\Xi}{\partial W_2}, \dots, \frac{\partial\Xi}{\partial W_p} \right). \quad (1.27)$$

Then, the weights are updated with the increment of

$$\Delta W_i = -\alpha \frac{\partial\Xi}{\partial W_i}, \quad (1.28)$$

for $i = 1, \dots, p$. The learning constant is defined by α . Thus, if ΔW_i increases, then $\partial\Xi/\partial W_i$ decreases, consequently the error is approaching to zero. The minimum of the error function is found when $\nabla\Xi = 0$ [37].

- **Backpropagation:** This is a learning method to train both, feedforward multi-layer ANNs and RNNs. Backpropagation uses gradient descent search to minimize the error function of the network. Next, the output error serves as

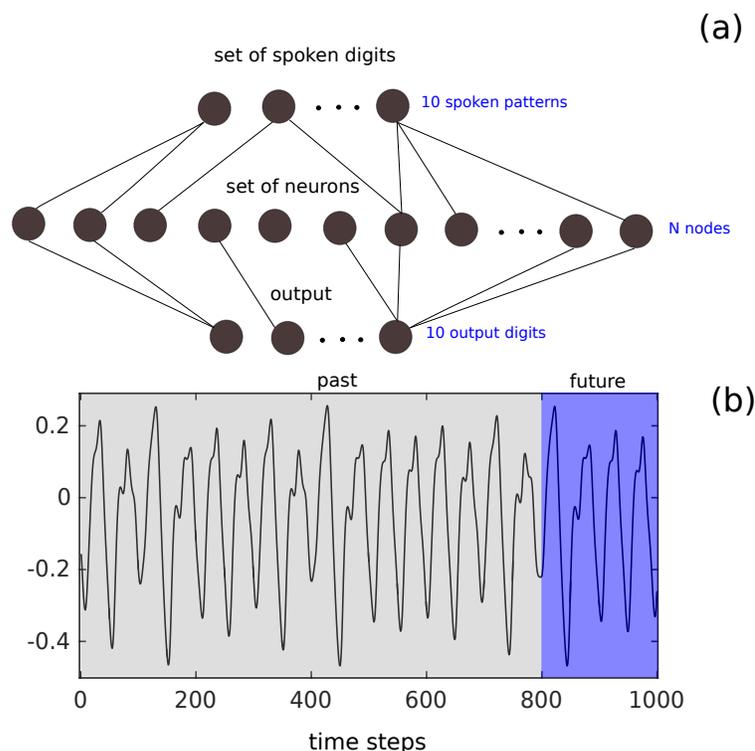


Figure 1.14: (a) Example of a network used for spoken digit recognition task, where the inputs are 500 patterns to be classified, with a network of N nodes and 10 outputs corresponding to the readable digits $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. (b) Signal from the MG system to be used for chaotic time series prediction. The first block in gray groups the past values used to train the ANN, the second block in blue contains the values to be predicted by the network.

input to the network via internal weights. The output errors are then backpropagated from the output to the internal layer. In RNNs, the backpropagation algorithm is applied considering that these networks are equivalent to feedforward ANNs with several layers, where each layer corresponds to a temporal iteration of the RNN [2, 3].

1.4.4 Benchmark tasks

As it was described already, ANNs are designed to solve some highly complex problems, that could not be solved previously in any other way. Some of the typical tasks that are addressed in the scientific literature are:

- **Classification:** The classification problem corresponds to the categorization of input data. The uncategorized input information is injected into the network, then the network has to separate and assign input subsets to different classes, that are the output units. As an example of a standard classification, the spoken digit recognition task uses a network to transcript the spoken language into readable digits, see Fig. 1.14(a).

- **Prediction:** This task corresponds to the estimation of future developments of a particular system, based on the knowledge of its past history. For example, in time series prediction the task corresponds to the estimation of value y_{n+T} (where n is the current time and T the prediction horizon) based on the knowledge of a past values set $\{y_1, y_2, y_3, \dots, y_n\}$. In Fig. 1.14(b) we show a signal from the MG system, Eq. (1.7) with $\tau_m = 17$, divided in two parts: a set of past values in gray, and a set of future values in blue. The task is to use a large set of past values to train the network, in order to obtain the future values in the blue section.

For prediction, neural networks exploit regularities within the training data for their learning procedures [43, 44]. Predictors from the connectionism framework include dynamic neural fields [45], ARMA-like model using Artificial Neural Network (ANN-ARMA) [46], feedforward [47], deep [48], and random neural networks [49, 50]. All receive information via injection through their input links. The manner to process information is different for each neural network, but in general the networks are capable to identify and mimic important regularities within the input information [43]. These regularities are repeating patterns present in the input information, and the networks are trained to identify and use them to solve prediction.

1.4.5 Reservoir computing

Reservoir computing is a field which includes the study of ANNs with a simplified design, where just the output layers are trained within a supervised learning rule. This trend started with echo state networks (ESNs) [5] and Liquid State Machines (LSMs) [6]. These two network architectures come from the fields of machine learning and computational neuroscience, respectively. More recently, a delay echo state network appeared as a novel computational machine. Such advantageous concept is found to be simple and implementable in hardware [7, 8]. Both approaches are introduced in the following.

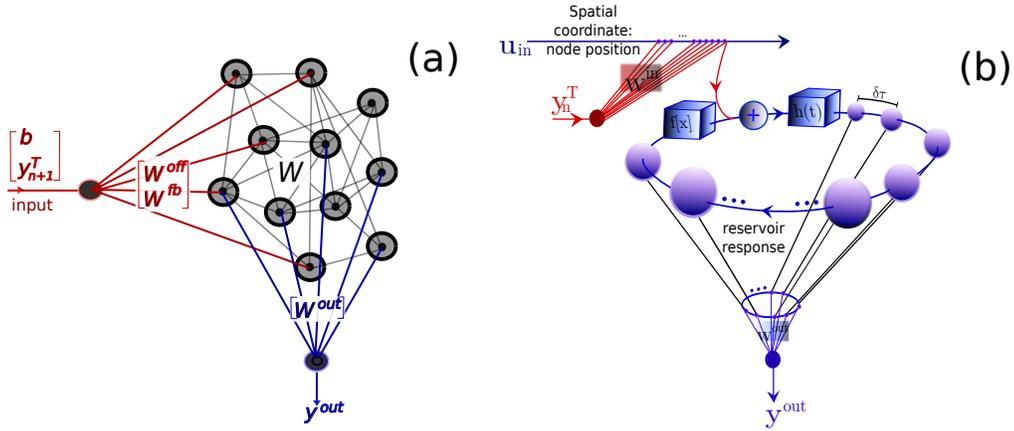


Figure 1.15: (a) Schematic diagram of Echo state networks, and (b) ring-like topology of a delay neural network.

1.4.6 Reservoir computing in spatio-temporal networks

A Random recurrent neural network (rRNN) is a popular model for investigating basic principles of information processing inside the human brain. In this model the synaptic neural links are Gaussian distributed [51, 52, 53, 54]. Although the brain's connectivity cannot be assumed to be fully random [55, 56], there is experimental evidence supporting the assumption that some parts of the brain are described by stochastic architectures. For example, in insects' olfactory systems the odour recognition process is performed by olfactory receptor neurons with structureless (random) synaptic connections [57, 58].

These networks consist of reservoirs of m neurons in state \mathbf{x}_n , internally connected in this case through a uniformly randomly distributed internal weights that are defined in a matrix W of dimensionality $m \times m$. The resulting randomly connected network is injected with input data $\{b, y_{n+1}^T\}$ according to random offset and teacher/input weights W^{off} and W^{fb} , respectively. The teacher/input weights W^{fb} have the label of feedback fb because, at a later stage, they will act as additional feedback. We normalize the largest eigenvalue of W to one. The time-discrete equation that governs the network is [5]

$$\mathbf{x}_{n+1} = f_{NL}(W \cdot \mathbf{x}_n + W^{off} \cdot b + W^{fb} \cdot y_{n+1}^T), \quad (1.29)$$

$$y_{n+1}^{out} = W^{out} \cdot \mathbf{x}_{n+1} \quad (1.30)$$

where $f_{NL}(\cdot)$ is a nonlinear sigmoid-like activation function, b a constant value which has the role of an offset. The network is trained via a supervised learning rule, where we estimate the output weights W^{out} through which we obtain the

network's output y^{out} . After training, the teacher signal y^T is replaced by the predicted output y^{out} , as already mentioned through the additional feedback involved with W^{fb} . The reservoir thus runs on its own to predict future values of the input signal, without any influence at that stage from any teacher (or input) signal y^T .

1.4.7 Delay reservoir computing

An often used mathematical description of the delay differential equation (DDE) for information processing is based on the Ikeda equation Eq. (1.20), that we re-write as follows [8, 59]

$$\tau \dot{x}(t) = -x(t) + f_{NL}[x(t - \tau_D) + \phi_0]. \quad (1.31)$$

It is also possible to re-formulate the Eq. (1.31) in a way where we can realize the spatio-temporal construction of the delay based reservoir [60, 61, 62]. For that, we write the integral convolution involving the impulse response $h(t)$ [63]. The definition of impulse response convolution of the filter h and the nonlinear function f_{NL} is

$$x(t) = (h * f_{NL})(t) \triangleq \int_{-\infty}^t h(t - \xi) f_{NL}[x(\xi - \tau_D)] d\xi, \quad (1.32)$$

where $h(\sigma)$ is the linear impulse response of the delay oscillator described by

$$h(t) = \begin{cases} e^{-t/\tau}/\tau; & t \geq 0 \\ 0; & t < 0 \end{cases} \quad (1.33)$$

In order to obtain the equivalent discrete-time system, we define the temporal variable as follows: $t = n\tau_D + \sigma_k$. The temporal position of each node is defined as $\sigma_k = (k - 1)\delta\tau$ for $k \in [1, m]$, $n \in \mathbb{N}$; with m as the total number of nodes that are filling one time delay interval: $\tau_D = m\delta\tau$. The variable of the system can then be written in a spatio-temporal form $x(t) \rightarrow x_n^k$. The integration variable of Eq. (1.32) can be defined in its discrete time (n) and continuous space (σ) form as follows: $\xi = n\tau_D + \sigma$. Here, $\sigma \in [0, \tau_D]$ corresponds to the short time scale of the dynamics, representing the spatial variable in the DDE. Now, we use the property of additivity of integration on intervals to write the time evolution of the system in Eq. (1.32)

$$x_n^k = \int_{-\infty}^{\sigma^{(1)}} h[(n - 1)\tau_D + \sigma_k - \xi] f_{NL}[x_{\xi - \tau_D}] d\xi + \quad (1.34)$$

$$\int_{\sigma^{(1)}}^{\sigma^{(2)}} h[(n - 1)\tau_D + \sigma_k - \xi] f_{NL}[x_{\xi - \tau_D}] d\xi, \quad (1.35)$$

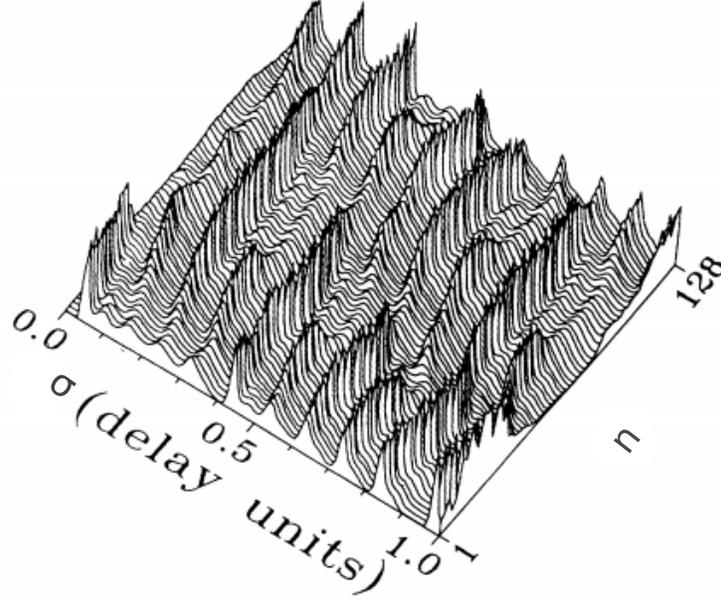


Figure 1.16: Spatio-temporal patterns in the $(\sigma - n)$ plane from Eq. (1.38) with $\alpha = 0$ [60].

where $\sigma^{(1)} = -\tau_D + \sigma_k$ and $\sigma^{(2)} = \sigma_k$ describe the evolution in a delay roundtrip (where in the continuous-time version $t \in [-\tau_D, 0]$). From this, in the second integral of last equation we substitute the new discrete-time variables, obtaining

$$x_n^k = \int_{-\infty}^{\sigma_k - \tau_D} h[(n-1)\tau_D + \sigma_k - \xi] f_{NL}[x_{\xi - \tau_D}] d\xi + \quad (1.36)$$

$$\int_{\sigma_k - \tau_D}^{\sigma_k} h(\sigma_k - \sigma) [x_{(n-1)\tau_D + \sigma}] d\sigma. \quad (1.37)$$

By substituting Eq. (1.32)

$$x_n^k = x_{n-1}^k + \int_{\sigma_k - \tau_D}^{\sigma_k} h(\sigma - \sigma_k) f_{NL}[x_{n-1}^\sigma + \phi_0] d\sigma, \quad (1.38)$$

where $x_n^\sigma = x_{n\tau_D + \sigma}$.

The scheme of a reservoir based on delay dynamical system is illustrated in Fig. 1.15(b). It consists of a ring-shaped neural network composed by virtual nodes separated by a distance $\delta\tau$, and distributed along the delay line. In

order to feed the reservoir through the input signal y_n^T , we use the technique of time division multiplexing

$$u_{in} = [W^{in} \cdot y^T] p_{\delta\tau}, \quad (1.39)$$

where W^{in} is the input weights vector composed by random numbers, and $p_{\delta\tau} = 1$ is a rectangular temporal windows with a width $\delta\tau$, which performs the temporal spacing between neighbours nodes, and n is the discrete-time variable of the reservoir. The input signal u_{in} multiplied by a constant α is added in the argument of f_{NL} in Eq. (1.38). The network is trained via regression, belonging to a supervised learning rule, where we estimate the output weights W^{out} through which we obtain the network's output y^{out} .

From the structure of Eq. (1.38), we can notice that it has a spatio-temporal representation of the reservoir, see Fig. 1.16. In comparison with the Strogatz-Kuramoto model [64] which has the same form as Eq. (1.38), here we find that $h(\sigma)$ is defining the number of neighbouring nodes (or oscillators) that are coupled, f_{NL} characterizes the type of nonlinear function involved in the coupling, and x_n^σ is the dynamical variable.

1.4.8 Regression

In this work, we use regression to estimate the output weights W^{out} :

Regression with singular value decomposition

The equation which relates a target y_n^T with a matrix containing all node responses of the network \mathbf{x}_n , is the following

$$y_n^T = W^{out} \cdot \mathbf{x}_n, \quad (1.40)$$

where W^{out} is the vector whose coefficients are needed to extract the computed output y_n^T of the reservoir from a linear combination of vector network state components of \mathbf{x}_n . To determine W^{out} it is necessary to minimizing the norm,

$$W_{op}^{out} = \min_{W^{out}} |W^{out} \cdot \mathbf{x}_n - y_n^T| \quad (1.41)$$

If $\mathbf{x}_n \neq 0$, and it is an asymmetric matrix, we can obtain its symmetric version by $\Lambda = \mathbf{x}_n^\dagger \mathbf{x}_n$, which is more convenient in order to determine the rank of a symmetric matrix, where $\text{rank}(\Lambda) = \text{rank}(\mathbf{x}_n)$. Then the Singular Value Decomposition (S.V.D.) of Λ is written as follows: $\Lambda = U\Sigma V^\dagger$, with

$$\Lambda^+ = V\Sigma^{-1}U^\dagger \quad (1.42)$$

as the Moore-Penrose inverse of Λ , where $(\cdot)^\dagger$ is the transpose of a matrix [65]. This definition is commonly used in solution of inverse problems, allowing to find the minimum of Eq. (1.41) for all possible W^{out} , by the decomposition of Λ into the product of three matrices. At this point, we want to construct a suitable basis for the information in Λ through linear mapping. For that, we define V as an orthogonal basis for the domain of this mapping, and U is an orthonormal bases for the co-domain. U is a transformation defined as $U \rightarrow \Lambda V$. Σ is a diagonal matrix which describes the scalar relationship between the vectors in V and in U . Then, the inverse of diagonal scaling Σ gives the correct relation between vectors in V and in U .

Ridge regression

The ridge regression is executed according to

$$W_{op}^{out} = M_{y^T} \cdot M_x^\dagger (M_x \cdot M_x^\dagger - \lambda \mathbf{I}_m)^{-1}, \quad (1.43)$$

where M_{y^T} and M_x are matrices containing information about target and node responses, respectively. λ is the regression parameter.

With the purpose to compare target y_n^T with the network's output values $y^{out} = W_{op}^{out} \cdot \mathbf{x}_n$, we introduce the *normalize mean squared error* (NMSE),

$$NMSE = \frac{1}{M} \frac{\sum_{t=1}^M (y_n^{out} - y_n^T)^2}{\delta^2(y_n^T)}. \quad (1.44)$$

where δ is the standard deviation.

1.5 Conclusion

In this chapter, we have exposed some of the main concepts and fundamental properties about dynamical systems theory. At first we focused on autonomous nonlinear dynamical systems which lead to chaotic solutions. These oscillators were also characterized by bifurcation phenomena as route to chaos, showing a variety of solutions such as fixed points and periodic trajectories. Also, we introduced some of the fields that dynamical systems theory can cover. As examples, we have shown systems from physiology, optics, chemistry and atmospheric sciences. Due to the impossibility of obtaining analytical solutions for chaotic systems, we also have reported several methods that help in the understanding and characterization of the dynamical complexity exhibited by such systems. Finally, we introduced neural networks, as models for driven

nonlinear dynamical systems. Their description allowed us to bring part of the theory of information processing in neural network into the context of dynamical systems. The aim of this manuscript, to be developed in the next chapters, is to exploit dynamical system analysis methods to possibly better explain the way how neural networks solve complex tasks.

Chapter 2

State-space prediction in Random Recurrent Neural Networks

In this chapter we address the question of how prediction of chaotic signals is solved by random neural networks, previously introduced in Sec. 1.4.4. In order to build a predictor for chaotic systems, most common techniques can be divided into the following groups: *(i)* linear and nonlinear regression models such as Moving-Average (MA), Autoregressive (AR), Autoregressive-Moving-Average (ARMA) [43], Multi Adaptive Regression Spline (MARS) [66], Support Vector Machine (SVM) [67], Least Square Estimation (LSE) [68], Non-linear Autoregressive with Exogenous inputs (NARX) [69], Wiener [70] and Volterra series [71]; *(ii)* state-space reconstruction-based techniques, which utilize interactions between internal degrees of freedom to infer the future [25, 43, 72, 73, 74], and *(iii)* connectionism framework, which uses neural networks.

In classical prediction theory, state-space reconstruction-based techniques allow to approximate the future if sufficient information about the system's previous states is available [25, 43, 72, 73]. However, in practice it is not always feasible to collect information of all state space dimensions of the chaotic system. In order to obtain a full state-space representation of the system to be predicted, most methods used by classical prediction theory are based on Whitney and Takens embedding theorems [21, 75]. Additionally, state-space reconstruction-based techniques for prediction require the exploration of the reconstructed attractor's neighborhood in state space, which attempts to complete missing information of the system's state history [25, 72, 73]. Such missing information is related to the practical impossibility to have a continuous-time state space representation of the data. Then, it is required to artificially increase the data resolution as much as possible if one wants to minimize the divergence between predicting model and actual chaotic system.

In the scientific literature about connectionism framework, there exist indications of possible embedding features in neural networks [76, 77]. However, the presence of this mechanism has so far been heuristically implied only. In the present chapter, we will demonstrate that classical state-space reconstruction-based models for prediction are indeed conceptually present in

random neural networks. Prediction in rRNNs therefore relies on similar principles. We demonstrate autonomous state-space reconstructions inside the high dimensional and complex neural networks' space, which allows these systems to employ a version of state-space reconstruction techniques. The random structure and high dimensionality of rRNNs open the possibility to use random projection theory to study such information embedding [78, 79].

2.1 State space reconstruction of chaotic systems

Inspired by random projections theory, we will identify task specific properties of random networks, analysing their suitability to embed one dimensional information in their high dimensional spaces. Since input information is multiplied by a randomly distributed weight matrix, the input signal is randomly projected onto a subspace of the network's high dimensional space. Consequently, there exists some spatial representation of the input time series. In the following, such spatial representation will be evaluated by using two exemplary input models: the Mackey-Glass (MG) and Ikeda systems.

2.1.1 Random projections theory

Random projections (RPs) theory is often employed as a dimensionality reduction tool for high dimensional datasets. Thus, an object defined in a particular space is embedded in fewer dimensions (projected) into a different space. Such embedding is carried out with a minimum structural damage to the original object. In the linear theory of RPs, RP preserves all interpoint distances with a relative error of $\epsilon > 0$ with high probability, where ϵ is a small constant, [80, 78, 79, 81]. The Johnson-Lindenstrauss lemma describes the structural preservation [80, 82],

Lemma 2. *For any $\epsilon > 0$, as a small constant. Let k be a positive integer such that $k = \mathcal{O}(\epsilon^{-2} \log S)$. Then for any set V of S points $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(S)}\} \in \mathbb{R}^d$, where d is large and $d \gg k$, there is a linear map $\mathcal{P} : \mathbb{R}^d \rightarrow \mathbb{R}^k$ such that for all $\mathbf{y}^{(i)}, \mathbf{y}^{(j)} \in V$,*

$$(1 - \epsilon) \|\mathbf{y}^{(i)} - \mathbf{y}^{(j)}\|^2 \leq \|\mathcal{P}(\mathbf{y}^{(i)}) - \mathcal{P}(\mathbf{y}^{(j)})\|^2 \leq (1 + \epsilon) \|\mathbf{y}^{(i)} - \mathbf{y}^{(j)}\|^2. \quad (2.1)$$

This lemma states that the distance between two consecutive points (interpoint distances) of the projected object does not lie out of the range $[(1 -$

$\epsilon), (1 + \epsilon)]$. Here, $\{\mathbf{y}^{(i)}, \mathbf{y}^{(j)}\}$ are two consecutive spatial points which come from the original object, and $\{\mathcal{P}(\mathbf{y}^{(i)}), \mathcal{P}(\mathbf{y}^{(j)})\}$ are two consecutive spatial points that come from the embedded object. Generally, $\mathcal{P}(\mathbf{y})$ is a projection that we obtain from the product between a $k \times d$ random matrix R with the incoming information \mathbf{y} , thus $\mathcal{P}(\mathbf{y}) = R\mathbf{y}$. Each spatial point \mathbf{y} and $\mathcal{P}(\mathbf{y})$, can be decomposed in terms of their components, $\mathbf{y} = (y_1, y_2, \dots, y_d)$ and $\mathcal{P}(\mathbf{y}) = (\mathcal{P}_1(\mathbf{y}), \mathcal{P}_2(\mathbf{y}), \dots, \mathcal{P}_k(\mathbf{y}))$. We measure the distance based on the euclidean norm between two consecutive points

$$\|\mathbf{y}^{(i)} - \mathbf{y}^{(j)}\| = \sqrt{[y_1^{(i)} - y_1^{(j)}]^2 + [y_2^{(i)} - y_2^{(j)}]^2 + \dots + [y_d^{(i)} - y_d^{(j)}]^2}; \quad (2.2)$$

and in the RP object

$$\|\mathcal{P}(\mathbf{y}^{(i)}) - \mathcal{P}(\mathbf{y}^{(j)})\| = \sqrt{[\mathcal{P}_1(\mathbf{y}^{(i)}) - \mathcal{P}_1(\mathbf{y}^{(j)})]^2 + [\mathcal{P}_2(\mathbf{y}^{(i)}) - \mathcal{P}_2(\mathbf{y}^{(j)})]^2 + \dots + [\mathcal{P}_k(\mathbf{y}^{(i)}) - \mathcal{P}_k(\mathbf{y}^{(j)})]^2}. \quad (2.3)$$

Here, we employ an extension of RPs theory to be used in the analysis of nonlinear systems, such as random neural networks. In such a situation the distortions of interpoint distances are not necessarily bounded to the interval $[(1 - \epsilon), (1 + \epsilon)]$. This is due to the fact that Eq. (2.1) requires the projection to be linear, i.e. $\mathcal{P}(\mathbf{y}^{(i)}) - \mathcal{P}(\mathbf{y}^{(j)}) = \mathcal{P}(\mathbf{y}^{(i)} - \mathbf{y}^{(j)})$. As a consequence, after a nonlinear random projection, interpoint distances are not restricted to the symmetric interval defined by ϵ . However, $\{\epsilon_1, \epsilon_2\}$ are proposed as constant values which define an asymmetric interval within which interpoint distances are bounded by $[(1 - \epsilon_1), (1 + \epsilon_2)]$ [83]. Consequently, let us introduce a more suitable estimate of the interpoint distances in the nonlinear approach,

Proposition 3. *For any positive constant values ϵ_1, ϵ_2 . Let V be a collection of S points $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(S)}\} \in \mathbb{R}^q$, with distances computed under L_2 norm. There is a map $\varphi : \mathbb{R}^q \rightarrow \mathbb{R}^h$, such that for all $\mathbf{y}^{(i)}, \mathbf{y}^{(j)} \in V$,*

$$(1 - \epsilon_1)\|\mathbf{y}^{(i)} - \mathbf{y}^{(j)}\| \leq \|\varphi(\mathbf{y}^{(i)}) - \varphi(\mathbf{y}^{(j)})\| \leq (1 + \epsilon_2)\|\mathbf{y}^{(i)} - \mathbf{y}^{(j)}\|. \quad (2.4)$$

This proposition states that after random nonlinear projections, the interpoint distances are bound to the range $[(1 - \epsilon_1), (1 + \epsilon_2)]$, where $\{\epsilon_1, \epsilon_2\}$ are arbitrary positive constant values that can be used to characterize random nonlinear projections. A modified version of RPs theory could be used as a potentially powerful tool to characterize possible attractor reconstruction in a random network. This theory may help to find out if distances between consecutive spatial datapoints are approximately preserved in the reconstructed

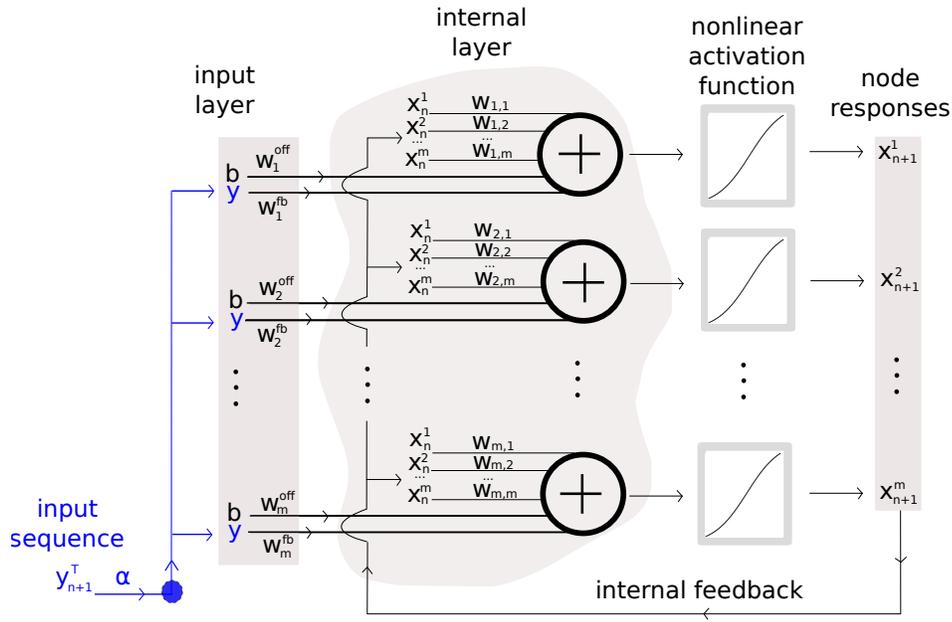


Figure 2.1: Explicit illustration of the rRNN diagram.

object [78, 79]. At this point, we wonder if RPs theory could provide us with the understanding of how the networks infer the behavior of the degrees of freedom associated to input dynamics. The random structure in rRNNs grants the use of RPs theory as basement to conceive a theory through which to analyse their possible attractor reconstruction capabilities.

2.1.2 Random Neural Networks

The classical rRNN layout from Sec. 1.4.6 is given in further detail in Fig. 2.1, which shows the explicit scheme illustrating the temporal flow of information received by each node. Nodes are represented by \oplus . Here, the task of each node is the addition of all the inputs $\{x_n^i, b, y_{n+1}^T\}$ according to random weights $\{w_{i,j}, w_i^{off}, w_i^{fb}\}$, and a nonlinear activation function. One of the effects of the nonlinear function is that it restricts the node responses to a defined region of the system's state space. The reservoir contains a collection of different nodes, where each node match with the simplified mathematical model of a neuron, see Sec. 1.4.1.

As introduced in Chapter 1 (Sec. 1.4.6), the time-discrete equation that

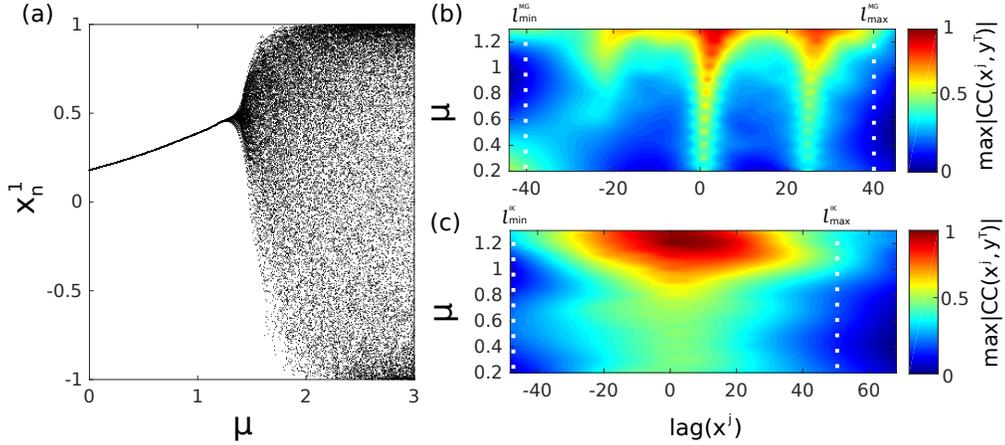


Figure 2.2: (a) Bifurcation diagram of the rRNN’s first node, under autonomous operation, with an increased feedback weight of the internal connectivity (equivalent to an increase of the so-called spectral radius of a network). (b,c) $|CC(x_j, y^T)|_{max}$ vs lag, between all node responses $\{x_n^i\}_i$ and the original data y^T , increasing μ for (b) MG and (c) Ikeda.

governs the network is now defined in more detail as follows [5]

$$\mathbf{x}_{n+1} = \tanh(\mu W \cdot \mathbf{x}_n + W^{off} \cdot b + W^{fb} \cdot y_{n+1}^T), \quad (2.5)$$

$$y_{n+1}^{out} = W^{out} \cdot \mathbf{x}_{n+1}; \quad (2.6)$$

where μ is a bifurcation parameter, also spectral radius. The uniformly random nature of the network’s structure allows the elements of the weights matrix W to serve as a basis where the data is going to be represented. However, since we cannot expect to have exclusively orthogonal directions as basis of the network’s state space from the random matrix, the exact dimensionality of the network is unknown. From here, we assume that the vectors defined in W are sufficiently close to orthogonal, hence they should be a good approximation of a basis [84]. The non-orthogonality of vectors in W might cause strong distortions in the dataset [85]. About the input sequence, the first value of y_{n+1}^T serves as initial condition in the first iteration, $n = 0$. This 1D input information y_{n+1}^T is at first mapped to the reservoir’s high dimensional space via W^{fb} . After an initial transient, the data is randomly projected onto a subspace of the reservoir state space, where it acquires its high-dimensional representation. Finally, the processed information \mathbf{x}_n is accumulated due to the recurrent nature of the network.

In practice, we construct a network with 1000×1000 random coefficients as

the reservoir layer, using the MATLAB routine `random`. Connection weights $w_{i,j}$ are distributed around zero. Hence 1000 neurons have 1000 synaptic connections to the rest of the network. We use 1000 nonzero connections, setting the connectivity of the random matrix equal to one. In the absence of input information ($\alpha = 0$), the autonomous dynamics of the network are scanned through the variation of the bifurcation parameter $\mu \in [0, 3]$. In Fig. 2.2(a), the bifurcation diagram of the rRNN shows the evolution of node 1 from steady state to chaotic dynamic. It is possible to observe two kinds of autonomous states of the rRNN: fixed point and chaos. The network changes suddenly to chaos for $\mu = 1.5$, from initial fixed point for ($0 \leq \mu \leq 1.4$).

Attractor embedding in rRNNs

We now consider the non-autonomous case of the network by injecting the input dataset y_{n+1}^T to it, which comes from the MG ($\tau_m = 17$) or Ikeda ($\tau_D = 20$) systems, see Secs. 1.2.2 and 1.2.3, respectively. Additionally, both time series have been shifted to oscillate around zero. The amplitude of injection is $\alpha = 0.8$ for MG and $\alpha = 0.1$ for Ikeda. Then, due to the fact that the input information is being randomly mapped in the network, we search for evidence of possible attractor reconstruction operation on it.

As described in Sec. 1.3.1, the inference of attractor dimensions is associated with independent observations that can be found using 1D datasets. The attractor reconstruction method based on Takens scheme, uses the autocorrelation function to reconstruct such coordinates, from the observation of a scalar variable from the dynamical system. Next, we now search for a possibility to identify the same independent observations inside our network's state space. For that, we make a similar analysis based on the maximum absolute value of the Cross-Correlation Analysis (CCA), $|CC(x_j, y^T)|_{max}$, between all node responses $\{x_n^i\}_i$ and the original input data y^T . For this purpose, we calculate the cross-correlation [86],

$$R_{xy}(l) = \begin{cases} \sum_{n=0}^{N-l-1} x_{(n+l)}^i y_n^{T*}, & l \geq 0 \\ R_{yx}^*(-l), & l < 0; \end{cases} \quad (2.7)$$

where l is the time lag, and N is the size of the dataset. This analysis was carried out using the Matlab routine `crosscorr`, and it will help to reveal if the network could infer independent observations in y^T , separating them by time lags. Each CCA is normalized to the $|CC(x^i, y^T)|_{max}$ when $\mu = 1.3$, which is the largest cross-correlation value of the whole analysis for all $\mu \in [0.2, 1.3]$.

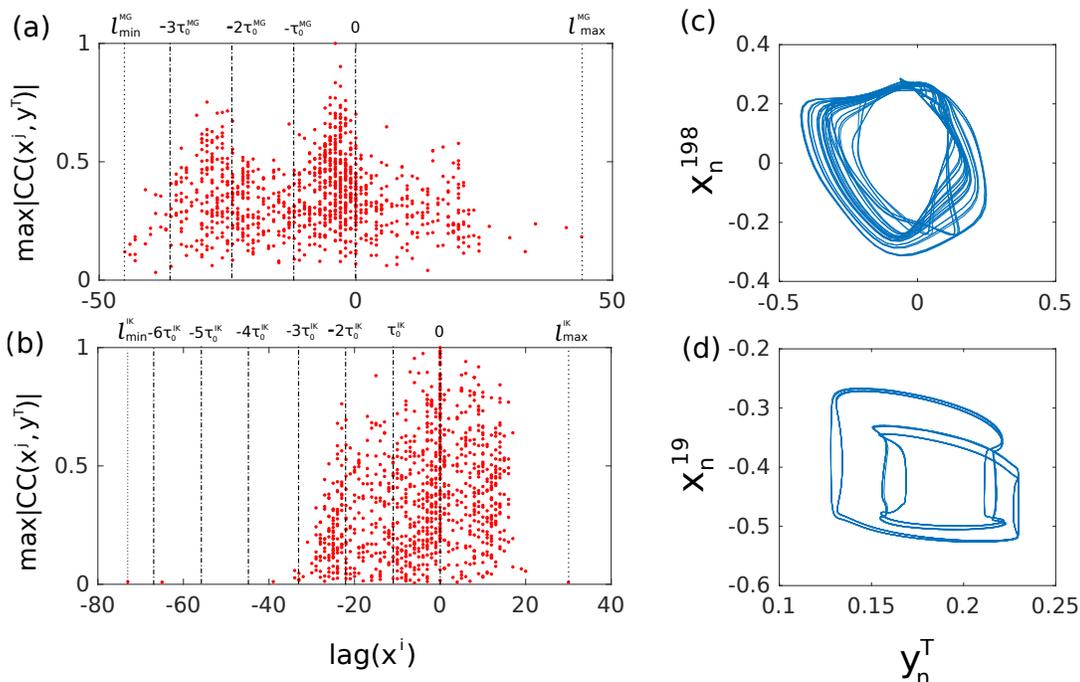


Figure 2.3: (a,b): CCA for MG (a) and Ikeda (b) for $\mu = 1.3$, in which each of the 1000 reservoir nodes is considered with a red dot located at its the maximum CC value vertically, and at its corresponding time lag for this maximum CC value horizontally. (c,d): Examples of 2D projections of embedded MG and Ikeda with nodes x_n^{126} and x_n^{19} , respectively.

Figure 2.2(b,c) shows the full CCA respect to μ for MG and Ikeda time series as the network's input, respectively. These analyses show that for small μ (typically $\mu < 0.8$) all $|CC(x^i, y^T)|_{max}$ values are located at lags within a few narrow temporal windows. It means that all nodes are time lagged at some few specific values. The increment of $\mu \geq 0.8$ corresponds to a redistribution of the location of nodes along many other lags. At $\mu = 1.3$, the distribution of nodes takes the widest range of lags among all μ . Such range is defined in Fig. 2.2(b,c) as $\{l_{\min}^{MG}, l_{\max}^{MG}\} = \{-40, 40\}$ and $\{l_{\min}^{IK}, l_{\max}^{IK}\} = \{-50, 50\}$ for MG and Ikeda, respectively. Our CCA stops at $\mu = 1.3$, due to the network bifurcates to a chaotic regime for higher values, see Fig. 2.2(a).

From the CCA, let us now focus on specific nodes whose $|CC(x^i, y^T)|_{max}$ are associated to time-lags revealed by the Takens embedding theorem applied to the input signals in Sec. 1.3.1. For $\tau_0^{MG} = 12$ (MG), and $\tau_0^{IK} = 11$ (Ikeda), any node of the reservoir revealing strong cross-correlation amplitudes at the

time lags $\{-3\tau_0^{MG}, -2\tau_0^{MG}, -\tau_0^{MG}, 0\}$ and $\{-6\tau_0^{IK}, -5\tau_0^{IK}, -4\tau_0^{IK}, -3\tau_0^{IK}, -2\tau_0^{IK}, 0\}$, can approximate any of the missing dimensions required to have a complete representation of the original state space of the complex input signal.

We analyse the case of $\mu = 1.3$, where the interval $[l_{min}, l_{max}]$ has the largest l_{min}, l_{max} lag limits in our study. Figure 2.3(a,b) shows $|CC(x^i, y^T)|_{max}$ as function of time-lag for MG and Ikeda, respectively. Here, the columns of values concentrated on lags $\{\pm 36, \pm 24, \pm 12, 0\}$ for the MG system, and $\{\pm 33, \pm 22, \pm 11, 0\}$ for Ikeda, correspond to most (or all) of the embedding delays required for Takens embedding the MG and Ikeda attractors. Therefore, these nodes approximate such an embedding, with which state-space like representations of the input sequences can be obtained. The rest of delay-coordinates are related to some embedding uncertainty which may belong to the lack of orthogonal directions in W , that usually cause strong distortions in the embedding procedure.

In Fig. 2.3(c,d) we show node responses x_n^{198} and x_n^{19} that are found to be good candidates for the embedding delay dimensions lagged at $l = \tau_0 = -12$ and -11 , for MG and Ikeda attractors respectively. These nodes were chosen due to the fact that they have the largest $|CC(x^i, y^T)|_{max}$ for the clusters of nodes lagged at $l = -12$ and $l = -11$. As it can be seen, the attractor reconstructions performed by random networks approximate 2D projections of the attractors shown by Fig. 1.9 of MG and Ikeda. This therefore illustrates that the missing coordinates of the input signal can be reconstructed by the rRNN.

2.2 Nearest neighbours

The chaotic systems used as input data for the random neural network are mathematically described by continuous-time differential equations. In the numerical simulation of any continuous-time system, one has to employ a time-discrete approximation such as e.g. Runge-Kutta method of 4th order or Euler method. However, such a discrete approximation always has some arbitrary resolution, which determines the accuracy of a numerical integration method. The higher the sampling resolution, the more accurate is the approximation of both, signal amplitude and temporal spacing distance. The distance between two consecutive points is therefore contaminated with an uncertainty of the order of the sampling resolution. As an illustration of a discrete representation, let us introduce a trajectory in state space generated by the discrete-time model, where each state is labelled with big black dots in Fig. 2.4(a), while

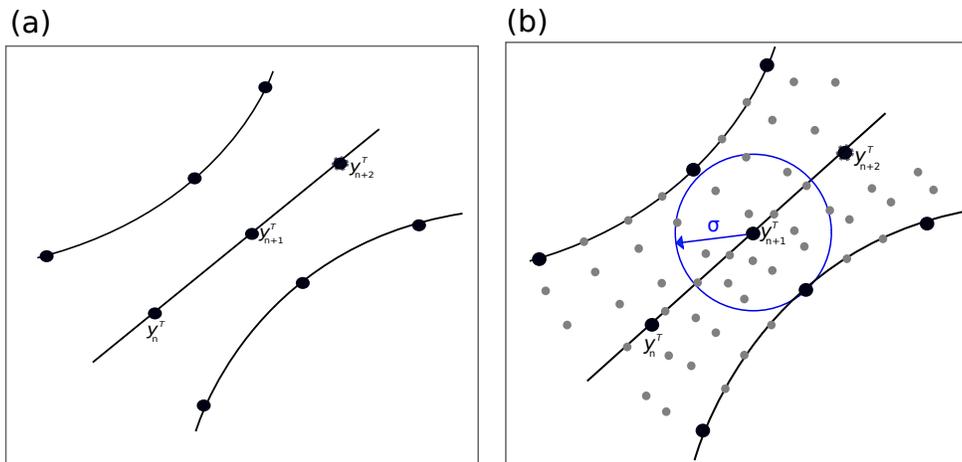


Figure 2.4: (a) Schematic illustration of the attractor's original trajectory. (b) Attractor's original trajectory (curve with big black dots), possible NN ensembles (curve with small gray dots), the region where the nearest neighbors can be found (blue circumference with radius σ) for an example of any state in the original trajectory [73].

the lines would correspond to the continuous-time trajectory of the original system.

If we want to enhance our current approximation of the continuous-time system, we may artificially increase the sampling resolution of the given trajectory in the state space. After such procedure, we get a new sampling resolution σ , see Fig. 2.4(b), which contains a set of new states within the trajectory's neighborhood. If we consider an arbitrary state \mathbf{y}_{n+1}^T , then all states within a neighborhood of radius σ centered in \mathbf{y}_{n+1}^T have to be considered a-priori equally good samples [25]. Then, the number of states around \mathbf{y}_{n+1}^T are all *nearest neighbors* (NN) to this state. The set of neighbors to state \mathbf{y}_{n+1}^T are illustrated with small gray dots forming a sampling cloud with radius σ . Such additional data points forming the sampling neighborhood are here created with a yet unspecified method.

One of the problem of predicting a chaotic time series originates from the fact that there is not any given mathematical model to describe the system we want to predict. The useful information available from a state space trajectory is therefore limited. Hence, it is not possible to deduce a-priori which route the system's trajectory will follow in the state space region where it is defined. The causal link between states of a given trajectory is missing. However, it is possible to approximate this causal relationship from features of the attractor.

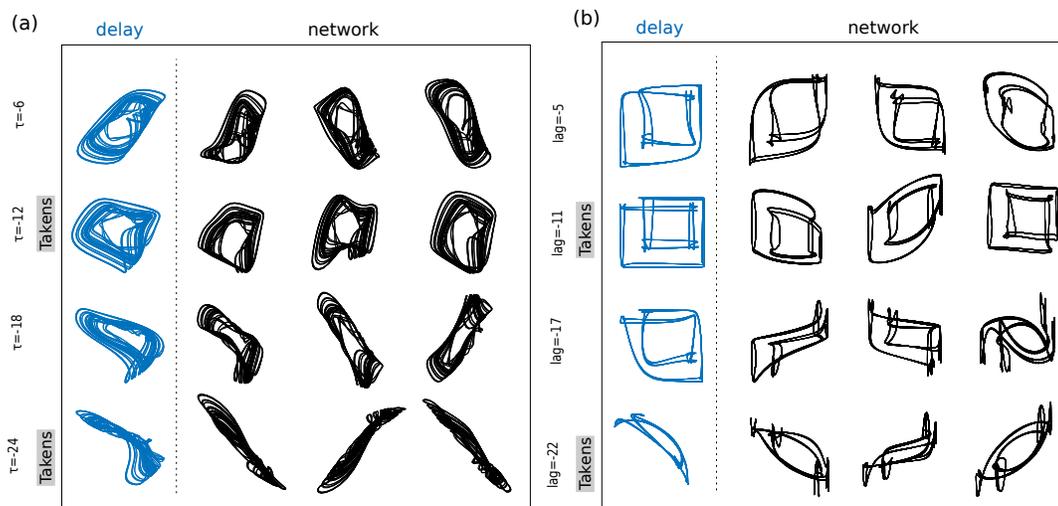


Figure 2.5: The first column shows reconstructed dimensions of the (a) MG and (b) Ikeda attractors reconstructed via time delay shifts. The following three columns in both cases show the embedding found inside rRNN space for $\mu = 1.3$.

The nearest neighbors approach provides a way to artificially increase the sampling resolution related with the location of all states. The increase of resolution allows to scan the vicinity of the attractor, with which one can construct more accurate approximations of pathways that the trajectory may follow in time.

2.2.1 Nearest neighbours from a rRNN

In the literature, numerous approaches to the generation of neighbors for time series prediction have been reported [43, 25, 72, 73, 74, 87, 88]. We will show that our random networks do create them as well. As discussed previously, RPs are capable to create several versions of the same geometrical object with small distortions. These distortions can be defined as modifications to the original attractor after their nonlinear projections in the network's space. They possibly can behave as neighbors sets to be used in our predictor of chaotic time series.

According to Fig. 2.3, 1D information is possibly reconstructed in the network's high dimensional space. In Fig. 2.5, we show evidence of how the rRNN's random subspace creates numerous modified versions of the original object. The first panels of Fig. 2.5 show the result from the 2D pro-

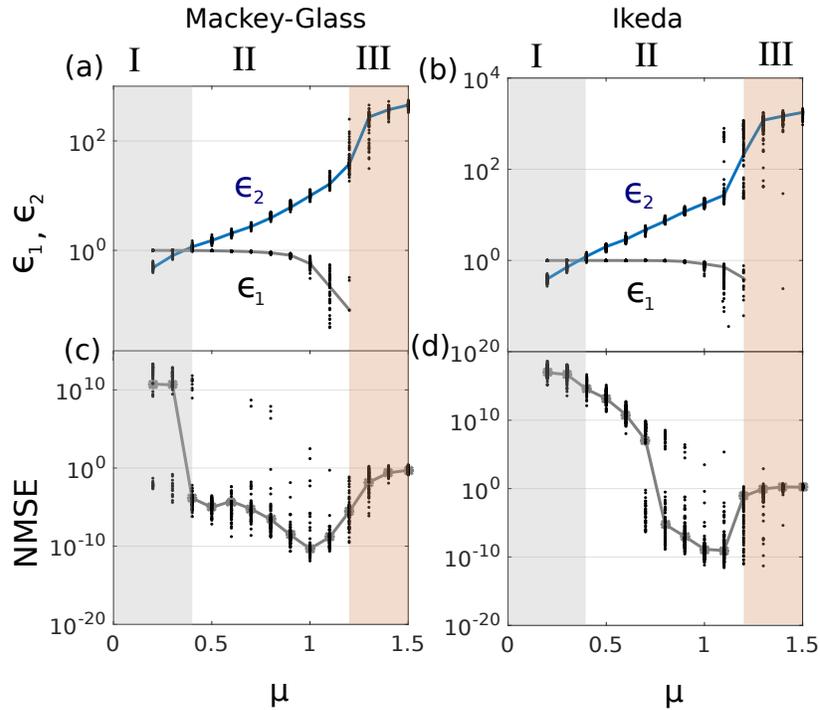


Figure 2.6: Maximum and minimum boundaries (ϵ_1, ϵ_2) of (a) MG and (b) Ikeda prediction. Sampling created by the rRNN is bound by these limits. The original attractor shape is preserved for $\mu < 1.2$, above which nonlinear projection loses validity. Prediction error (NMSE) of MG and Ikeda are shown in (c) and (d), respectively. Embedding by the rRNN requires sampling to span a phase space volume until the future value, corresponding to $\epsilon_1 > 1$. At $\epsilon_2 \gtrsim 1$, we therefore find a strong transition. Below prediction diverges, beyond the rRNN approximates the target attractor.

jections from the Takens scheme, reconstructed by using delay-reconstruction $\{-24, -18, -12, -6\}$ for the MG, and $\{-22, -17, -11, -5\}$ for the Ikeda system. The attractors reconstructed by network nodes with maximum cross-correlation values at lags $\{-18, -6\}$ for MG and $\{-17, -5\}$ for Ikeda, do not belong to the set of original Takens delay-coordinates. However, such additional delay-coordinates allowed us to geometrically visualize the large variety of nodes lagged at other values in the network's space.

We can now define the conditions to apply Proposition 3, in order to understand how the neighbors are created inside the rRNN's state space. The first step is to define two arbitrary consecutive points $\{\mathbf{y}^{(i)}, \mathbf{y}^{(j)}\}$ in the Takens

space of the input sequence. For that, we write points $\{\mathbf{y}^{(i)}, \mathbf{y}^{(j)}\}$ as states $\{\mathbf{y}_n^T, \mathbf{y}_{n+1}^T\} \in \mathbb{R}^q$, where $q = M = 4$ for MG and $q = M = 7$ for Ikeda. Each state in Takens space is described by q delay-coordinates,

$$\mathbf{y}_n^T = (y_n^T, y_{n+\tau_0}^T, \dots, y_{n+(M-1)\tau_0}^T), \quad (2.8)$$

$$\mathbf{y}_{n+1}^T = (y_{n+1}^T, y_{(n+1)+\tau_0}^T, \dots, y_{(n+1)+(M-1)\tau_0}^T). \quad (2.9)$$

The second step is to define the corresponding two arbitrary consecutive points $\{\varphi(\mathbf{y}^{(i)}), \varphi(\mathbf{y}^{(j)})\}$ in the rRNN state space. For that we write points $\{\varphi(\mathbf{y}^{(i)}), \varphi(\mathbf{y}^{(j)})\}$ as states $\{\varphi(\mathbf{y}_n^T), \varphi(\mathbf{y}_{n+1}^T)\} \in \mathbb{R}^h$, where h depends on μ . The value of h is determined from the CCA, where we approximately assign the mapped objects dimensionality to the number of elements found in the interval $[l_{min}, l_{max}]$ for each μ , see Fig. 2.2(b,c). In order to construct those points, we use all the delay-coordinates provided by the network, i.e. the full range $[l_{min}, l_{max}]$ for each value of μ , as follows

$$\varphi(\mathbf{y}_n^T) = [\varphi_{l_1}(\mathbf{y}_n^T), \varphi_{l_2}(\mathbf{y}_n^T), \dots, \varphi_{l_h}(\mathbf{y}_n^T)], \quad (2.10)$$

$$\varphi(\mathbf{y}_{n+1}^T) = [\varphi_{l_1}(\mathbf{y}_{n+1}^T), \varphi_{l_2}(\mathbf{y}_{n+1}^T), \dots, \varphi_{l_h}(\mathbf{y}_{n+1}^T)]; \quad (2.11)$$

where $\{\varphi_{l_1}(\mathbf{y}_n^T), \varphi_{l_2}(\mathbf{y}_n^T), \dots\}$ are node responses lagged at $[l_{min}, l_{max}]$. The size of the interval $[l_{min}, l_{max}]$ depends on the value of μ , as it was shown by Fig. 2.2(b,c), where we find a broader distribution of delay-coordinates for higher values of μ .

The interpoint distances $\|\mathbf{y}^{(i)} - \mathbf{y}^{(j)}\|$ and $\|\varphi(\mathbf{y}^{(i)}) - \varphi(\mathbf{y}^{(j)})\|$, here corresponding to the *interstate distances* $\|\mathbf{y}_{n+1}^T - \mathbf{y}_n^T\|$ and $\|\varphi(\mathbf{y}_{n+1}^T) - \varphi(\mathbf{y}_n^T)\|$, have to be bounded in the interval $[(1 - \epsilon_1), (1 + \epsilon_2)]$ according to

$$\frac{\|\varphi(\mathbf{y}_{n+1}^T) - \varphi(\mathbf{y}_n^T)\|}{\|\mathbf{y}_{n+1}^T - \mathbf{y}_n^T\|} \in [(1 - \epsilon_1), (1 + \epsilon_2)]. \quad (2.12)$$

Under these conditions, we can claim that the transformation by the rRNN agrees with a nonlinear RP. Estimating limits $\{\epsilon_1, \epsilon_2\}$ requires to find the inferior ϵ_{min} and superior ϵ_{max} interstate distance limits:

$$\frac{\|\varphi(\mathbf{y}_{n+1}^T) - \varphi(\mathbf{y}_n^T)\|_{min}}{\|\mathbf{y}_{n+1}^T - \mathbf{y}_n^T\|} = \epsilon_{min}; \quad \frac{\|\varphi(\mathbf{y}_{n+1}^T) - \varphi(\mathbf{y}_n^T)\|_{max}}{\|\mathbf{y}_{n+1}^T - \mathbf{y}_n^T\|} = \epsilon_{max}, \quad (2.13)$$

where ϵ_1 and ϵ_2 are calculated by isolating these constants from $\epsilon_{min} = (1 - \epsilon_1)$, and $\epsilon_{max} = (1 + \epsilon_2)$. These limits contain information about the minimum and

maximum distortions that we can find in order to get the best neighbors in the rRNN. $\|\varphi(\mathbf{y}_{n+1}^T) - \varphi(\mathbf{y}_n^T)\|_{min}$ and $\|\varphi(\mathbf{y}_{n+1}^T) - \varphi(\mathbf{y}_n^T)\|_{max}$ are calculated by using Euclidean distance under minimum and maximum norms,

$$\|\varphi(\mathbf{y}_{n+1}^T) - \varphi(\mathbf{y}_n^T)\|_{min} = \left(\sum_{l_g=l_{min}}^{l_{max}} [\varphi_{l_g}(\mathbf{y}_{n+1}^T) - \varphi_{l_g}(\mathbf{y}_n^T)]_{min}^2 \right)^{1/2}, \quad (2.14)$$

$$\|\varphi(\mathbf{y}_{n+1}^T) - \varphi(\mathbf{y}_n^T)\|_{max} = \left(\sum_{l_g=l_{min}}^{l_{max}} [\varphi_{l_g}(\mathbf{y}_{n+1}^T) - \varphi_{l_g}(\mathbf{y}_n^T)]_{max}^2 \right)^{1/2}, \quad (2.15)$$

where $\varphi_{l_g}(\mathbf{y}_n^T)$ are node responses lagged at $l_g \in [l_{min}, l_{max}]$, $\forall g = 1, 2, \dots, h$.

Here, we therefore identify the smallest and largest distances $[\varphi_{l_g}(\mathbf{y}_{n+1}^T) - \varphi_{l_g}(\mathbf{y}_n^T)]_{min,max}$ along each delay coordinate. Finally, we determine $\|\mathbf{y}_{n+1}^T - \mathbf{y}_n^T\|$ via

$$\|\mathbf{y}_{n+1}^T - \mathbf{y}_n^T\| = \sqrt{(y_{n+1}^T - y_n^T)^2 + \dots + (y_{(n+1)+(M-1)\tau_0}^T - y_{n+(M-1)\tau_0}^T)^2}. \quad (2.16)$$

Based on Eq. (2.13), Eq. (2.14), Eq. (2.15) and Eq. (2.16) we can now obtain limits ϵ_1 and ϵ_2 . We repeat this procedure for each $\mu \in \{0.1, 0.2, \dots, 1.3\}$. Figure 2.6(a,b) shows the estimation of $\{\epsilon_1(\mu), \epsilon_2(\mu)\}$ for the rRNN at different μ for 100 different realizations of random matrix W , where the lines correspond to the average of the statistical distribution. For each realization, we train the network via regression (supervised learning) according to procedure described in Sec. 1.4.8, using the MATLAB routine `pinv`, and determining W^{out} for each value of μ . Once trained, the input signal is replaced by the network's own output $y_{n+1}^T = y_{n+1}^{out}$ [5]. Next, we let the rRNN run freely for prediction horizons where the network needs to fully embed the input system in order to perform long-term predictions. These horizons correspond to 150 time steps in MG (~ 9 delays of the MG equation) and in Ikeda (~ 7 delays of the Ikeda equation) systems. When we evaluate the prediction performances via the estimation of the normalised mean squared error (NMSE), Eq. (1.44), we see in Fig. 2.6(c,d) that $\epsilon_2(\mu)$ begins to be greater than one at the point where the rRNN can perform long-term predictions. At this transition, found at $\mu \leq 0.4$ for MG and Ikeda prediction, some realizations of W result in divergent results (NMSE $\gtrsim 10^9$), and some others in good long-term predictions (NMSE $\lesssim 10^{-4}$). This transition is caused by the network when it starts exploring the projected object's neighborhood.

The increase of interpoint distances limits $\{\epsilon_1(\mu), \epsilon_2(\mu)\}$ with μ can be explained in an intuitive way through the graphic representation of the limits

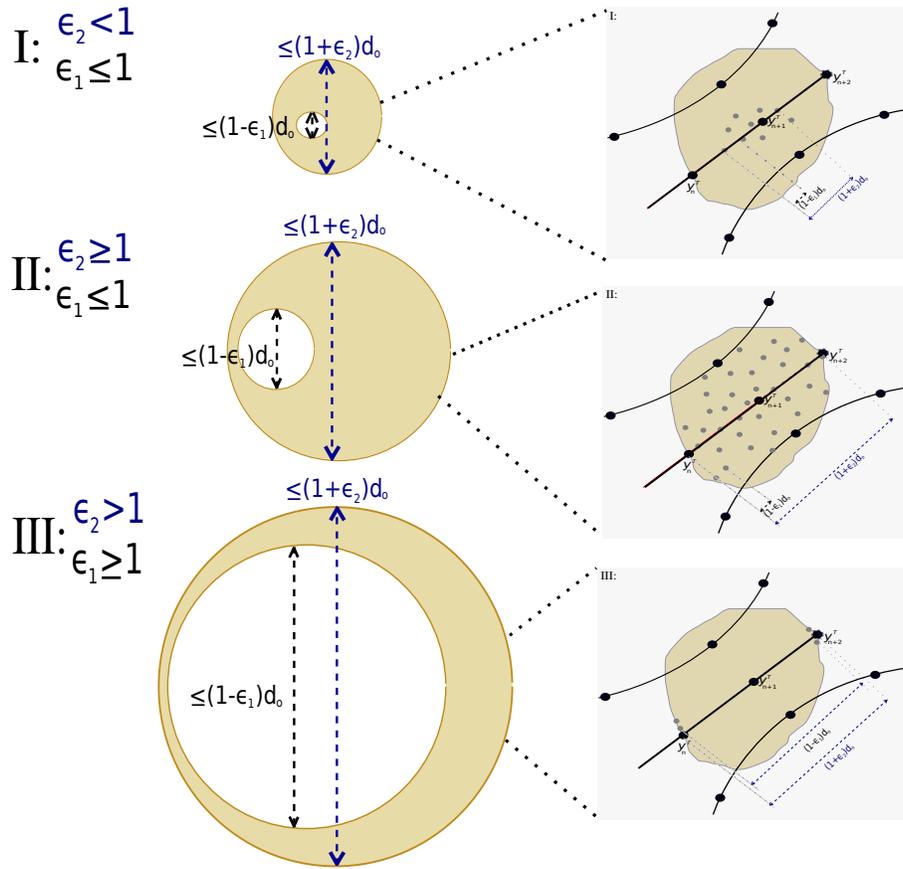


Figure 2.7: Illustrative scheme of the evolution of $\{\epsilon_1, \epsilon_2\}$ as function of μ , and an example of how are the NN distributed when the long term prediction starts.

shown by Fig. 2.6(a,b). In Fig. 2.7 we illustrate three general cases (I,II,III), based on the region created between the balls whose diameter are defined by smallest $(1 - \epsilon_1)d_o$ and largest $(1 + \epsilon_2)d_o$ distances, found in the network's space. Here, d_o defines the interpoint distances in the original state space of the trajectory to be predicted. The first case (I) corresponds to neighbors which form a dense cloud of samples distributed around the original trajectory when $\epsilon_1 \lesssim 1$ and $\epsilon_2 < 1$. In the scheme on the right panel, we then introduce a more explicit schematic interpretation of the distribution of neighbors: Let us represent a given dynamical system's state space by a trajectory composed by a curve connecting the big black dots, which are the acquired data points. Gray dots are network's neighbors states to the original trajectory. The neighbor

samples that the network created insufficiently scan the neighborhood of state y_{n+1}^T . Since such example is extensible to all other states, then the network cannot predict.

The next case (II) includes the values of μ where $\epsilon_2 \geq 1$, and long term prediction starts to be executed (iff the dimensionality property is fulfilled). At this point, the neighbours contribute to start long term prediction when $\epsilon_2 \geq 1$. Then, the maximum interpoint distance possible inside the rRNN's space is twice the interpoint distance of the original trajectory. It means that the neighbors of exemplary state \mathbf{y}_{n+1}^T provide information of the region where the future state is. So as it can be seen on the right side, there is a sufficient scanning of the attractor's vicinity, so the random network can use the projected objects to solve prediction.

As μ is increasing, the area of this intermediate region is decreasing until it becomes very small. This last case (III) appears for $\mu > 1.3$, where all distances are much larger than the original one d_o . Then there is absence of attractor's neighborhood scanning. This feature corresponds to the stretching action of the rRNN onto the injected information. The information is folded in a limited region of the network's state space due to the limits of the nonlinear function. The projected object is stretched in a shape preserving way until the network reaches chaotic behavior. In fact, according to the bifurcation diagram in Fig. 2.2(a), the trajectory inside the reservoir's space might be completely distorted by the autonomous rRNN dynamics typically found for $\mu \gtrsim 1.4$, which are chaotic in this regime. Consequently, the network's folding property distorts the data in such a way that the projected interstate distances $\|\varphi(\mathbf{y}_{n+1}^T) - \varphi(\mathbf{y}_n^T)\|$ cannot be mapped back onto their original distance, $\|\mathbf{y}_{n+1}^T - \mathbf{y}_n^T\|$. Therefore, ϵ_1 becomes undefined, meaning that specific information about the structure of the embedded trajectory is lost. A confirmation of these interpretations is directly provided from results reported in Fig. 2.6(c,d).

2.3 Takens-inspired rRNN

In this section, we present a modified version of the classical random neural network for time series prediction. As it was described previously, the delay-coordinates in rRNNs potentially coincide with the coordinates from Takens Embedding Theorem for $\mu > 0.4$, although with some embedding uncertainty. Under such conditions, we can find a broad spectrum of delay-coordinates where not all of them can be related with the Takens scheme. In order to build a network capable to specifically generate Takens-delay-coordinates, we

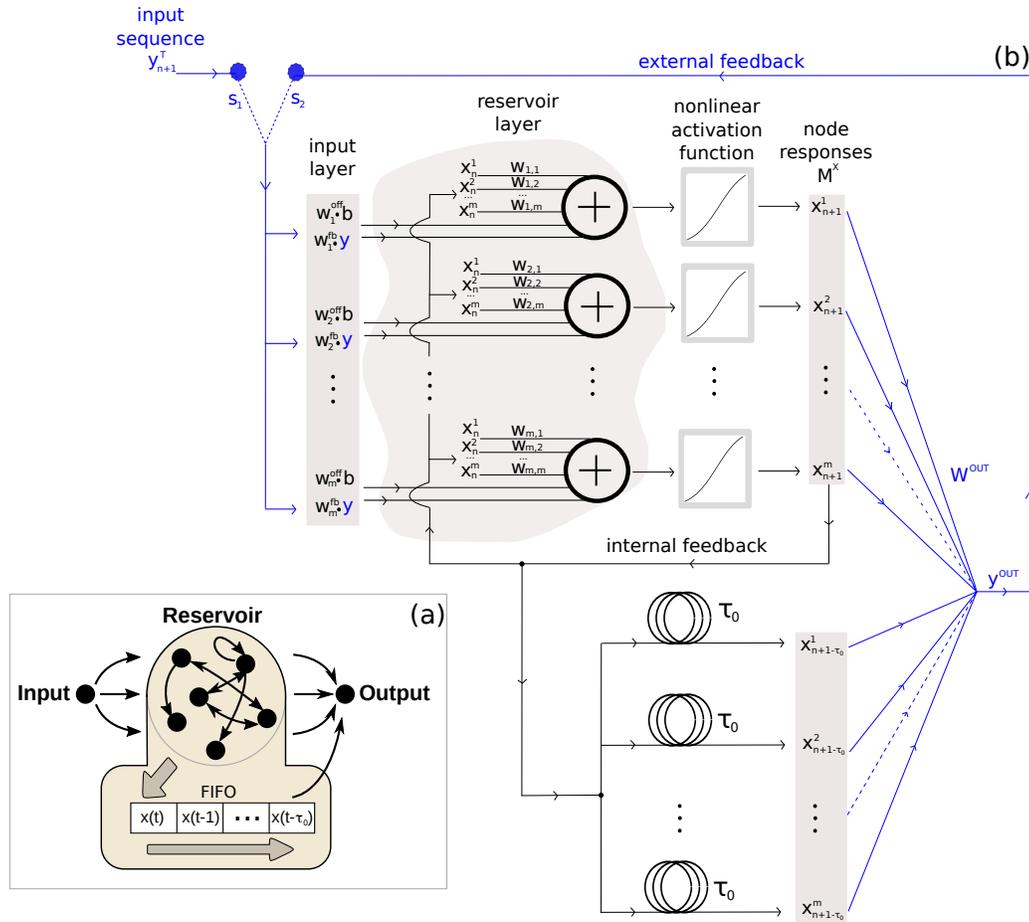


Figure 2.8: (a) Schematic illustration of an TrRNN. Information enters the system via the input, a recurrently connected network forms a neural network. Based on our theory, we propose a simplistic extension to the system via an external first-in first-out (FIFO) memory. (b) Explicit illustration of the TrRNN diagram. The network has an internal connectivity W , a τ_0 delay layer is added to the scheme. A readout state y^{out} is created via the readout weight matrix W^{out} . Switch S either connects the network to teacher signal y^T (S_1 training), or to y^{out} (S_2 testing). In both cases the driving signals are connected to the network via input connectivity W^{fb} .

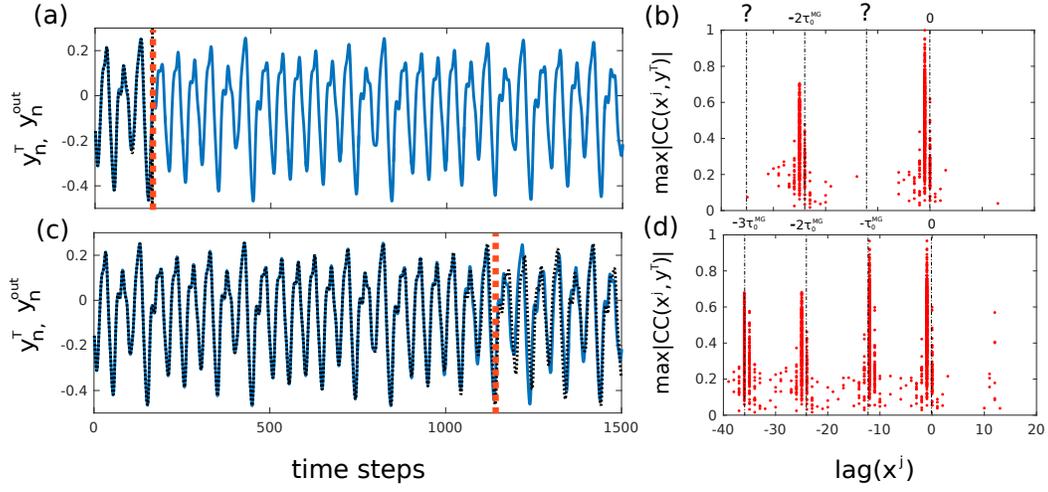


Figure 2.9: y_n^T and y_n^{out} sequences for (a) rRNN and (c) TrRNN to predict 1000 steps with $\mu = 0.2$. CCA showing the delay-like dimensions available in the system for (b) rRNN and (d) TrRNN. In (a,c) the red dashed line highlight the points where prediction starts diverging from the target.

include an additional delayed term in the classical rRNN equation,

$$\mathbf{x}_{n+1} = f_{NL}(\mu W \cdot \mathbf{x}_n + W^{off} \cdot b + W^{fb} \cdot y_{n+1}^T), \quad (2.17)$$

$$y_{n+1}^{out} = W^{out} \cdot (\mathbf{x}_{n+1}, \mathbf{x}_{n+1-\tau_0}) \quad (2.18)$$

where $\mathbf{x}_{n+1-\tau_0}$ is the delay term in the output layer. In Fig. 2.8, the delay term is schematically represented by a nonvolatile external memory. All elements of the node responses layer have been time shifted by Takens embedding delay to $(n + 1 - \tau_0)$. W^{out} weights have been assigned as well to the Takens shifted node-states in the training step, and the time shifted layer contribute to the rRNN's internal state via the external feedback (S_2 testing).

Afterwards, we evaluate our Takens rRNN (TrRNN) for prediction of 1500 steps into the future of the MG system. In Fig. 2.9(a) we present the predicted time series by using a rRNN with $\mu = 0.2$. As it can be seen, y_n^{out} (dashed curve) does not match with the original sequence y_n^T (solid curve) for long-term prediction. In fact, rRNN's output and target strongly diverge, which also resulted in the large prediction error shown in Fig. 2.6.

The CCA in Fig. 2.9(b) shows that this bad prediction performance is related to the non-existence of the required temporal embedding dimensions. Only just a couple of delay-coordinates were found ($l = 0$ and $l = 2\tau_0$) if $\tau_0 = -12$, and therefore the standard rRNN cannot embed the full MG attractor. Once the

τ_0 delay term is incorporated in the output layer, we find that the TrRNN is capable to perform long-term predictions for $\mu = 0.2$, see Fig. 2.9(c). Here, the dimensions needed to embed the MG attractor are present in the TrRNN as revealed by the additional dimensions within the CCA analysis located at $l = \{0, \tau_0, 2\tau_0, 3\tau_0\}$ for $\tau_0 = -12$, Fig. 2.9(d). This confirms our postulates about Takens-like embedding and random recurrent networks. Our new scheme efficiently uses the rRNN as it reorganizes all the neighbours only to Takens dimensions.

2.4 Application: control an arrhythmic neuronal model

In the following, we study the stabilization of a system which models the firing behavior of a noise-driven neuron. It consists in the FitzHugh-Nagumo (FHN) neuronal model [89, 90],

$$\epsilon_{fhn} \frac{dv(t)}{dt} = v(t)[v(t) - g_{fhn}][1 - v(t)] - w + I + \xi(t), \quad (2.19)$$

$$\frac{dw(t)}{dt} = v(t) - D_{fhn}w(t) - h_{fhn}, \quad (2.20)$$

where $v(t)$ and $w(t)$ are voltage and recovery variables. $I = 0.3$ is a tonic activation signal, ξ is Gaussian white noise with zero mean and standard deviation ~ 0.02 , $\epsilon_{fhn} = 0.005$, $g_{fhn} = 0.5$, $D_{fhn} = 1.0$, and $h_{fhn} = 0.15$. These equations have been solved by the Euler-Maruyama algorithm for stochastic differential equation's integration, see Fig. 2.10(a). In its resting state, the neuron's membrane potential is nearly negative. Once the membrane voltage $v(t)$ is sufficiently depolarized through an external stimuli, the neuron spikes due to the rise of action potentials. In the case of a large system of neurons, the spiking activity can be considered a random event, due to the fact that perturbations from the other neurons arrive at random [91, 92]. The time between consecutive spikes are defined as *interspike intervals* (ISIs). These random ISIs are shown by Fig. 2.10(b), where no clear pattern in the ISIs' behavior is apparent.

We aim to control this random spiking behavior of the FHN neuronal model by proportional perturbation feedback (PPF) [93] and our networks. PPF method consists in the application of perturbations to locate the system's unstable fixed point onto a stable manifold [93, 94]. This method is used to fit instabilities in the FHN neuronal model. In our case, the goal of using the

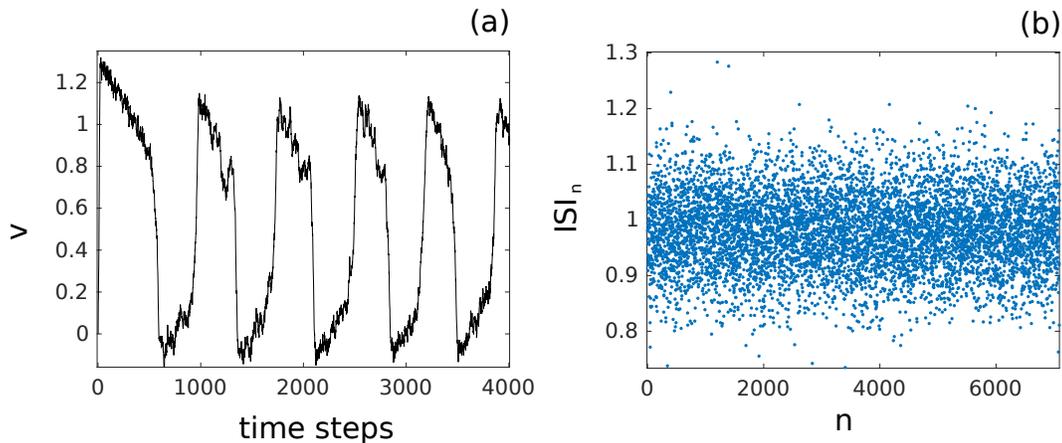


Figure 2.10: (a) Time series of the voltage from the FHN system. (b) Random ISI activity.

PPF method is to apply an external stimuli to trigger spiking and reduce the degree of chaos. The PPF algorithm first estimates the position of the fixed points according to [95],

$$\tau_{\star} = \frac{c_{fhn}}{1 - a_{fhn} - b_{fhn}}, \quad (2.21)$$

where parameters $a_{fhn}, b_{fhn}, c_{fhn}$ are calculated from the system:

$$\begin{aligned} \tau_{k+1} &= a_{fhn}\tau_k + b_{fhn}\tau_{k-1} + c_{fhn}, \\ \tau_{k+2} &= a_{fhn}\tau_{k+1} + b_{fhn}\tau_k, \\ \tau_{k+3} &= a_{fhn}\tau_{k+2} + b_{fhn}\tau_{k+1}, \\ \tau_{k+4} &= a_{fhn}\tau_{k+3} + b_{fhn}\tau_{k+2}. \end{aligned} \quad (2.22)$$

The τ_i 's are a small collection of ISIs. In this analysis $c_{fhn} \equiv 1$. The system is near a fixed point at time k if the pair of points (τ_k, τ_{k+1}) and (τ_{k+1}, τ_{k+2}) form a pattern that fits a line of shallow slope in $[-1, 1]$. In Fig. 2.11(a) it can be seen an illustrative scheme of the dynamics in the fixed point neighborhood. There are several planes where the dynamic is approximately linear. Each plane consists in a manifold that corresponds to a set of ISI points $\{\tau_k, \tau_{k+1}, \tau_{k+2}\}$ including different asymptotic trajectories to the line of shallow slope. The existence of various planes illustrates the instability of the trajectories near the fixed point. Following this approach we identify a sequence of τ_{\star_i} , repeating this analysis with all ISI points, and selecting sets of 5 ISI points each time.

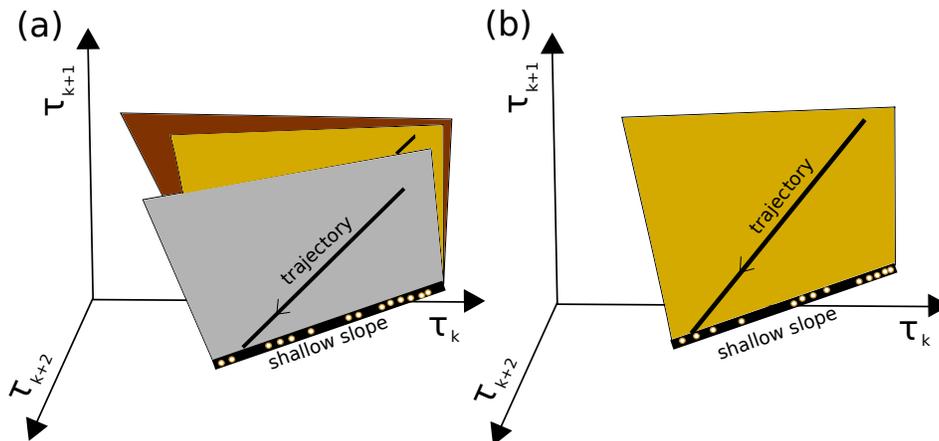


Figure 2.11: Schematic illustration of (a) the dynamic near the fixed point, where each plane consists in a manifold of ISI points; (b) a unique plane corresponding to the stable manifold when control is activated.

Then, we determine the histogram of τ_{\star_i} , and estimate the location of the fixed point τ_{\star} to be the sharpest peak's position defined as C_h . We collect all τ_{\star_i} located in an open ball whose center is C_h and radius ϵ , i.e. $|C_h - \tau_{\star_i}| < \epsilon$. Next, we collect an amount of j ISI values within the ball and estimate a_{fhn} and b_{fhn} .

From this point we start with the design of a control line which specify a unique plane in Fig. 2.11(a). At first, we use the pair $\{a_{fhn}, b_{fhn}\}$ to calculate:

$$B_{1,2} = \mathbb{R} \left\{ \frac{a_{fhn} \pm \sqrt{a_{fhn}^2 + 4b_{fhn}}}{2} \right\},$$

where $B_{1,2} = \{B_1, B_2\}$. Then, we collect all $B_{1,2}^j$ values and determine an amount of l values within the open ball $|a_{fhn} - B_{1,2}^j| > 0$. In this step we estimate the histograms of the sets of l -values found under the previous condition, and determine their maximum values with centers D_1 and D_2 . These lasts values are the parameters required to build the control equation for our system,

$$\tau_{max} = A\tau_k + C, \quad (2.23)$$

where $C_{fhn} = (1 - A_{fhn})C_h$. If $D_1 > D_2$, then $A_{fhn} = D_1$, otherwise $A_{fhn} = D_2$. By using this control line we can stabilize the fixed point, due to it locates the fixed point in a stable manifold, i.e. in a single plane in Fig. 2.11(b).

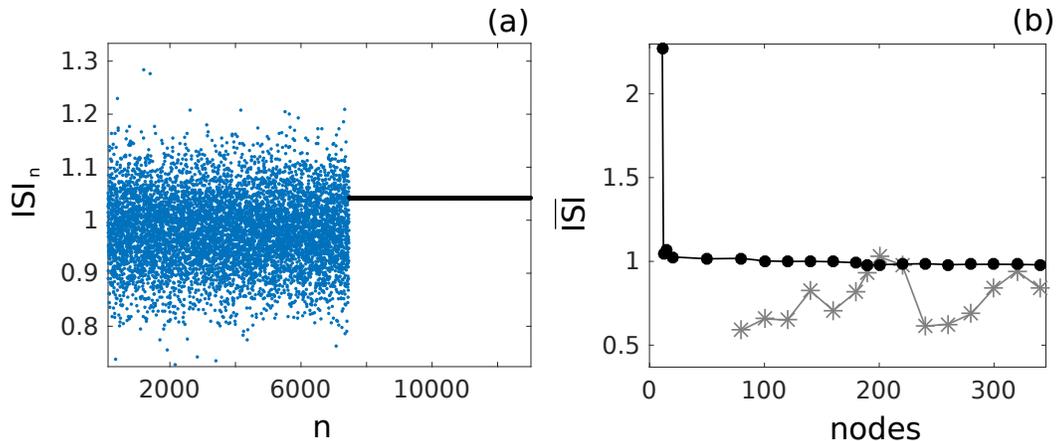


Figure 2.12: (a) Interspike intervals (ISI) of a arrhythmic excitable system comparable to a heart. Stabilization of the system based on our Takens rRNN (TrRNN). (b) Comparison between the stabilized mean of the TrRNN (black curve with dots) and a classical rRNN (gray curve with stars). The TrRNN requires 15 times less nodes, simultaneously achieving superior performance.

Once the control is activated for $n = 7465$ in our simulation, the control signal Eq. (2.23) is injected via I through a train of pulses which take discrete values $\{0.2, 5.2\}$. The pulses are injected when the current ISI exceeds τ_{max} .

In our approach, the past information provided by the voltage $v(t)$ is used to determine two things: (i) the control parameters A_{fhn} , C_{fhn} , and (ii) training parameters for rRNN ($\mu = 1.3$) and TrRNN ($\mu = 0.2$, and $\tau_0 = 166$ obtained from the Takens scheme) to predict future values of $v(t)$. The predicted $v(t)$ is used to calculate the full control signal with which we stabilize the neuron's spiking activity. To train the networks, we inject 1×10^5 values and we let the network run free for other 4×10^6 steps, allowing us to stabilize 5619 ISI points. We then evaluate the quality of the stabilization for networks of different size ranging from 11 to 340 nodes.

In Fig. 2.12(a) we show the average and normalized ISI when stabilization is implemented based on either the TrRNN. As it can be seen, the TrRNN can control the random ISI from $n = 7465$. Figure 2.12(b) shows the outcome of the full analysis using classical rRNN and TrRNN, where the mean value of ISI is calculated as function of the number of nodes used to build any network. The TrRNN starts inferring the inner dynamic of the FHN system just with an architecture consisting in 12 nodes, providing predicted information to build the control signal required to stabilize ISI. In contrast, the classical rRNN does

not predict at all until it has a minimum amount of 80 nodes in its architecture, but performance remains poor in comparison with TrRNN. For 190 nodes the rRNN starts predicting the dynamic of the FHN system, allowing the control signal to fully stabilize ISI. Yet, for more than 200 nodes the good performance still can fluctuate, even significantly dropping again. This is an indicator that in general the stabilization via the classical rRNN is not robust. Furthermore with the TrRNN, we can reduce the number of nodes to 15 times less than the classical rRNN. Our TrRNN has a reduction of all connections by a factor of 224 with respect to the classical rRNN.

2.5 Conclusion

We have introduced a novel methodology, demonstrating how prediction is achieved by rRNNs. Our theory firmly integrates the field into a nonlinear dynamical systems treatment. rRNNs and prediction can consequently be described via a common methodology. Quantifying measures such as the introduced memory related correlation analysis or the sampling neighborhoods are therefore interpretable. We significantly extend the toolkit previously available for Neural Network analysis.

Our scheme has numerous practical implications. The most direct is the motivation and development of new training strategies addressing temporal problems in rRNNs in general. Furthermore, we already identify a strategy of how hybrid-computers can efficiently be exploited for prediction by a priori defining external memory access rules. Finally, our work partially removes the black-box property of neural networks for prediction, possibly giving translational insight into how such tasks can be solved in comparable systems.

The limitation of our approach are related with the fact that the parameters $\{\epsilon_1, \epsilon_2\}$ provide us information about the maximum and minimum possible boundaries found in the rRNN. However, they do not specify how neighbors are actually distributed in the network. That is to say, we cannot know if the neighbors are truly randomly distributed once created, or if they form clusters around a particular state, etc. Consequently, a more deep investigation about the distribution and functionality as a whole or in sub-assemblies is expected for future works.

Chapter 3

Dynamical complexity and computation in neural networks beyond their fixed point

Despite the fact that interactions inside rRNNs are governed according to random interconnections, these networks can still achieve highly coherent collective behavior [49, 55, 96, 97]. Under such conditions they can experience phase synchronized dynamics, which have been identified to play an important role in biological memory processes [98], neural communication [99, 100, 101] and plasticity [102, 103, 104, 105]. Phase synchronization therefore plays a crucial role in biological neural networks for information processing. An example of several phase synchronized signals with zero lag are shown in Fig. 3.1(a). As for their biological prototype, regular spatio-temporal patterns of self-organization can also be found in homogeneous [106] as well as in heterogeneous artificial neural networks [50, 107]. If all signals are zero lag phase synchronized, then the spatio-temporal patterns have the structure shown by Fig. 3.1(b). The here discussed rRNNs are heterogeneous networks which have been employed for the modelling of biological neural networks. Conveniently, their dynamical state can be tuned via a single parameter, which typically results in a bifurcation phenomena as a route to chaos [41, 108, 109]. Furthermore, this transition in their dynamical properties might have a spatio-temporal impact due to the collective network's dynamical evolution towards disordered dynamical regimes. One way to analyse such spatio-temporal patterns is through the measure of the synchronization between nodes.

Besides their function as model systems in biological neuroscience, rRNNs have been widely studied in the machine learning community due to their excellent computational properties. In rRNNs, special attention was given to computation at the transition from a steady state to chaotic dynamics, defined as the edge of chaos of the dynamical system. Essential for solving complex tasks, operating a network at the edge of chaos ensures a high susceptibility to perturbations [40, 41, 108, 109, 110]. Typically, when the network reaches the regime where it experiences dynamics in the absence of stimulation by external data, its spontaneous dynamics are considered a nuisance as they disrupt the

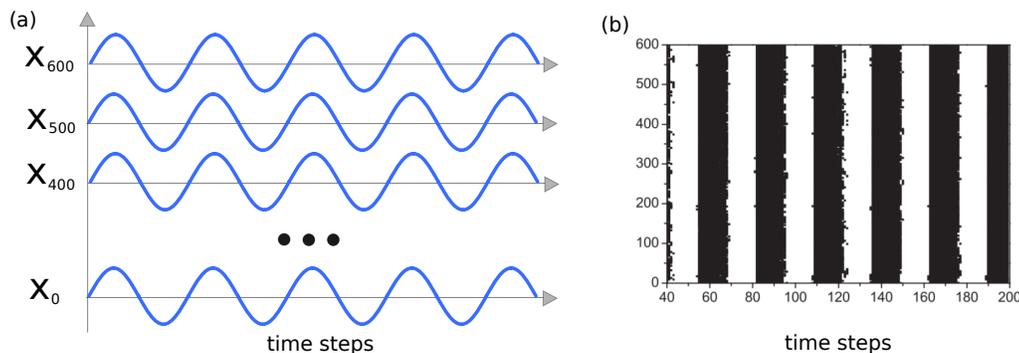


Figure 3.1: Illustrative examples of (a) phase synchronized signals with zero lag and (b) regular spatio-temporal patterns [107].

causality between input and network state. Nevertheless, information processing might not depend on the regularity of individual node dynamics. In other words: details of an autonomous rRNN's dynamical state should be of minor importance, as long as the network as a whole can preserve the content of injected information. Global dynamical properties and their use for computation therefore deserve a closer inspection.

Here, we study a rRNN predicting a chaotic time series. Motivated by the impact of spatio-temporal dynamical properties, we particularly focus on beyond fixed point operation. We employ a variation to classical rRNNs [5, 41, 108, 109] by using nodes with a sinusoidal activation function, with feedback parameter outside the argument of the activation function. A broad range of autonomous dynamics are the consequence, among which we most importantly find multiple non-fixed point states with surprisingly high computational performance. We show that for computation based on such non-fixed point states, phase synchronization and bifurcation point play essential roles in information processing. The underlying mechanisms are analysed based on the mutual information between the rRNN and the input time signal, as well as the rRNN's maximal Lyapunov exponent. Our choice of system is directly motivated by its randomness: we can exclude structural modifications induced by learning being the cause behind phase synchronization.

3.1 Random recurrent neural networks

As for the networks introduced in Chapter 1, our rRNN consists of m nodes in state \mathbf{x}_n , internally connected via a random, uniformly-distributed internal

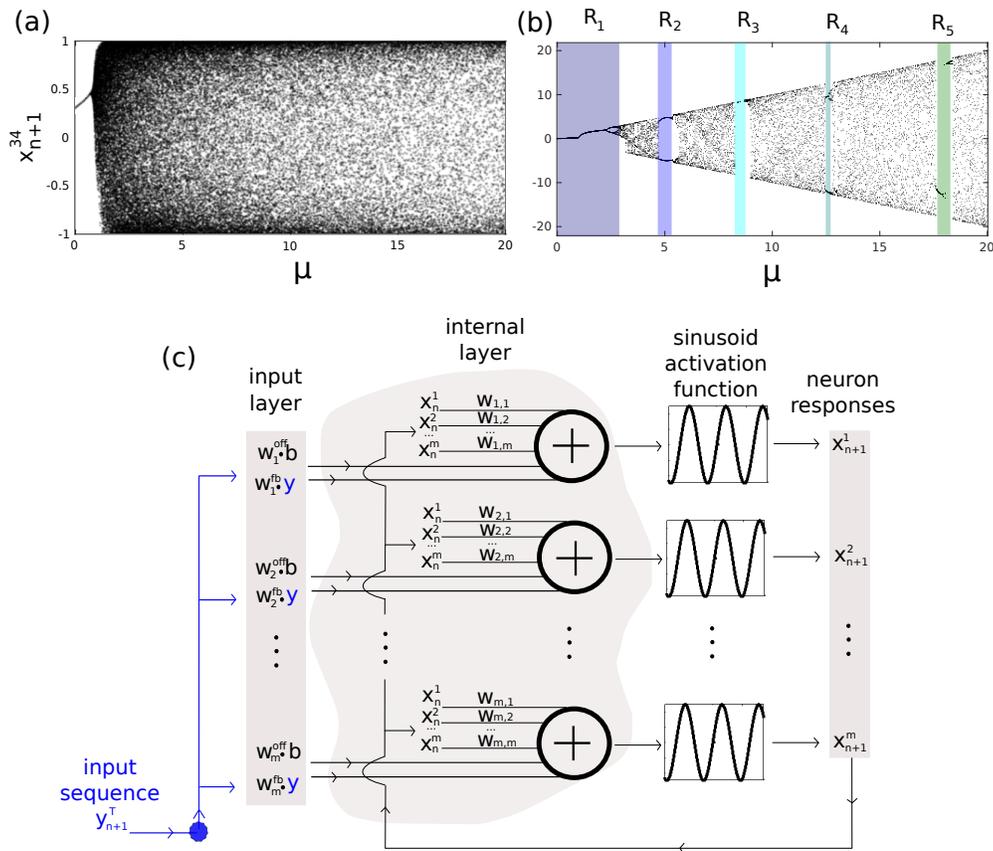


Figure 3.2: (a) Bifurcation diagram of node 34 (x_{n+1}^{34}) for the classical rRNN with $F_{NL} = \tanh(\cdot)$. (c) Schematic illustration of the rRNN structure with $F_{NL} = \sin(\cdot)$. (b) Bifurcation diagram of node 34 (x_{n+1}^{34}) for a rRNN with $F_{NL} = \sin(\cdot)$, with regions of non-chaotic behavior: R_1 , R_2 , R_3 , R_4 , and R_5 .

weight matrix W of dimensionality $m \times m$. The resulting random networks have a temporal evolution governed by

$$\mathbf{x}_{n+1} = \mu F_{NL}(W \cdot \mathbf{x}_n + W^{off} \cdot b + W^{fb} \cdot \alpha y_{n+1}^T), \quad (3.1)$$

where $\{W, W^{off}, W^{fb}\}$ are random weights that have been generated using the MATLAB routine `sprand`, $b = 0.2$ is a constant offset, $\alpha \cdot y_{n+1}^T$ is the input signal, where α the input scaling and μ is the feedback amplification. In this first case, the connectivity matrix W is constructed with 500×500 random, Gaussian-distributed coefficients in $[-0.5, 0.5]$, from a matrix with connectivity one. Typically, the nonlinear activation function is $F_{NL} = \tanh(\cdot)$

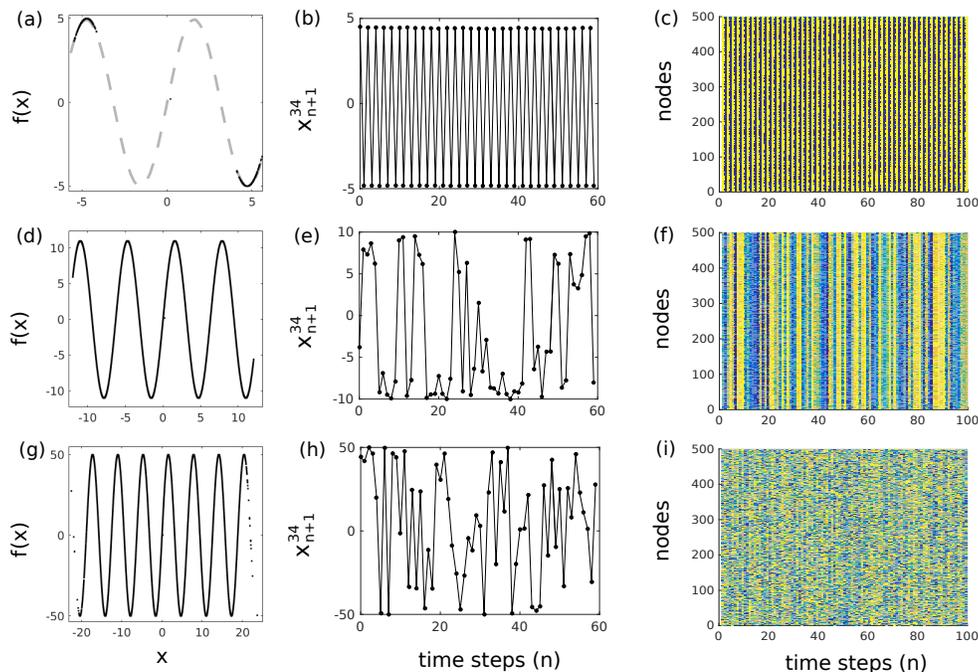


Figure 3.3: Nonlinear response of nodes with $f(x) = \sin(x)$ (left column), time series recorded for node x_{n+1}^{34} (center column), and spatio-temporal evolution of the rRNN (right column): (a-c) $\mu = 5$, (d-f) $\mu = 10$, and (g-i) $\mu = 50$. All dynamics obtained for the autonomous system ($\alpha = 0$).

[5]. However, for $\mu \geq 1.4$ and the hyperbolic function as F_{NL} , there are no other steady state windows that can be used for information processing, i.e. the network's dynamic falls directly into a chaotic regime. This fact is revealed by the bifurcation diagram shown by Fig. 3.2(a) where μ in $[0, 20]$, for a randomly chosen node, i.e. node 34 (x_{n+1}^{34}) of the autonomous system ($\alpha = 0$).

In order to obtain a neural network experiencing multiple windows of regular dynamics in its bifurcation diagram, we modify the standard rRNN scheme, now using a sinusoidal activation function $F_{NL} = \sin(\cdot)$. The schematic diagram of the whole network with $F_{NL} = \sin(\cdot)$ is shown by Fig. 3.2(b). Nodes (symbol \oplus) add and nonlinearly transform all inputs $\{\mathbf{x}_n, b, y_{n+1}^T\}$ according to random weights $\{W, W^{off}, W^{fb}\}$. Otherwise we use the same number of nodes and connectivity, however the coefficients of W are now randomly distributed in $[0, 1]$. The bifurcation diagram, for node 34 and $\alpha = 0$ (autonomous operation) shows a typical evolution from steady state to chaotic dynamic through periodic oscillations, and even with periodic windows between chaotic regimes,

see Fig. 3.2(c). Non-chaotic regimes can be found in several regions: R_1 , R_2 , R_3 , R_4 , and R_5 , for $\mu \in [0.1, 2.8]$, $\mu \in [4.8, 5.4]$, $\mu \in [8.5, 8.9]$, $\mu \in [12.4, 12.7]$, and $\mu \in [17.6, 18.4]$, respectively. These regions are part of the multistability present in this particular network, consequence of the sinusoidal nonlinear function.

Autonomous rRNN dynamics are shown by Fig 3.3 for different values of the bifurcation parameter μ . The left column shows the functional input-output relationship of Eq. (3.1) with $F_{NL} = \sin(\cdot)$ for the randomly selected node 34. The central column displays time series from the same node, referred to as local dynamics, while the right column shows the dynamical state of the full rRNN. For $\mu = 5$ the rRNN's state is symmetrically concentrated along the nonlinear function's extrema, see panel (a) of Fig. 3.3. The resulting dynamics of x_{n+1}^{34} and the full networks state \mathbf{x}_{n+1} is shown in Fig 3.3(b,c), respectively. Autonomous dynamics of x_{n+1}^{34} are therefore periodic, and according to Fig 3.3(c) such local periodic dynamics appear to strongly synchronize across the rRNN. The result is a constant phase relation within the entire network. When increasing bifurcation parameter μ to 10, dynamics span an increasing number of the nonlinear function's periods, see Fig 3.3(d). The consequence is a local dynamical state with considerably higher complexity, see Fig 3.3(e). Yet, synchronization between individual nodes proves to be robust. According to Fig 3.3(f) regular spatial patterns are still present and synchronization across the network is preserved despite the chaotic dynamics of individual nodes. Finally, for further increasing μ to 50, dynamics spanning up to 14 extrema (Fig 3.3(g)) result in hyper-chaotic node responses, Fig 3.3(h). As illustrated in Fig 3.3(i), only for such large bifurcation values the regular spatio-temporal structures across the network have vanished and synchronization is lost.

3.2 Mitigating autonomous dynamics by learning

All previously discussed dynamical properties have been exclusively obtained in the absence of an external stimulus. However, as an information processing system, the rRNN realizes computation on the basis of rich dynamical responses to external, i.e. sensory input. We therefore activate the rRNN's input by setting $\alpha \neq 0$ and investigate dynamics when fed by a chaotic MG time series. We add an output layer that provides the computational result according to

$$y_{n+1}^{out} = \tanh(W_{op}^{out} \cdot \mathbf{x}_{n+1}). \quad (3.2)$$

A divergence of the system for large feedback amplification μ is avoided by artificially limiting the range of y^{out} through a hyperbolic tangent in the output. The output weight vector W^{out} receives all node responses \mathbf{x} and is calculated according to a supervised learning rule based on a teacher/target signal αy_{n+1}^T . Here, the training target is equivalent to the input signal, shifted by a single time step. We use 2000 values from the Mackey-Glass (MG) system (introduced in Sec. 1.2.2 with $\tau_m = 17$) and $\alpha = 0.8$. For the system driven by the teacher signal we estimate the optimal output weight vector W_{op}^{out}

$$W_{op}^{out} = \min_{W^{out}} \|\tanh(W^{out} \cdot \mathbf{x}_{n+1}) - y_{n+1}^T\| \quad (3.3)$$

via its pseudo-inverse according to singular value decomposition. Equation (3.3) therefore minimizes the error between output $\tanh(W^{out} \cdot \mathbf{x}_{n+1})$ and teacher y_{n+1}^T . As training error measure we use the NMSE between output y_{n+1}^{out} and target signal y_{n+1}^T .

Once trained, the input signal is replaced by the network's own output $y_{n+1}^T = y_{n+1}^{out}$ in Eq. (3.1), and the system autonomously approximates dynamics learned from the teacher system, here the MG sequence. Computational performance is determined after a free evolution of 35 time steps, twice the time-delay of the MG sequence ($\tau_m = 17$). In Fig 3.4(a) the prediction performance is characterized by NMSE that is shown for $0 < \mu \leq 10$. At each μ we repeated the previously introduced training procedure. The optimal performance (NMSE= 5.5×10^{-4}) is found for a very narrow regime around $\mu = 0.9$, which comes at no surprise as it corresponds to the well known edge of chaos condition. However, additionally we identify multiple broad regions of acceptable performance with a prediction error of roughly NMSE $\sim 10^{-2}$. A comparison to the rRNN's bifurcation diagram of Fig 3.2(b) demonstrates that these regions of acceptable performance directly correspond to regular regions R_1 , R_2 and R_3 . Regimes R_4 and R_5 are not treated in our analysis, since small perturbations result in their stabilization, driving the network instantaneously into the next chaotic regime. In all other regions the error is orders of magnitude higher, therefore not corresponding to any prediction ability of the rRNN operated under such conditions.

As previously introduced, in this self-driven operation mode the rRNN's output becomes its own input. If not suppressed, perturbation-like autonomous rRNN dynamics can therefore freely propagate through the system due to this recurrent rRNN input/output relationship. Furthermore, in a worst case such architecture could result in resonances between output and response, with uncertain consequences for prediction. Yet, the deviation between target and prediction, measured by the NMSE, is up to three orders of magnitude

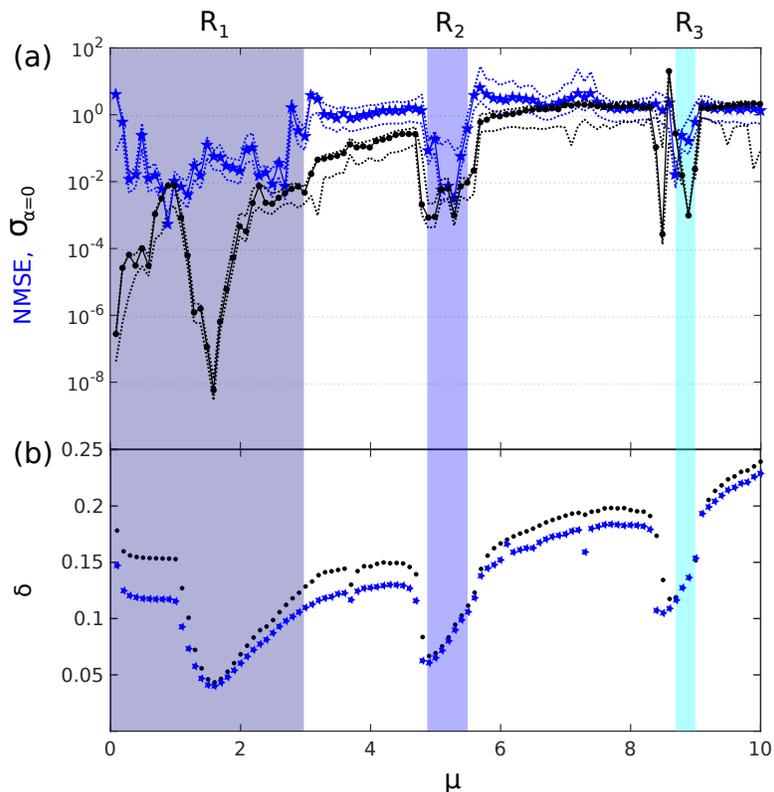


Figure 3.4: (a) Average NMSE as a function of μ (blue stars), and the noise-like standard deviation in rRNN output $\sigma_{\alpha=0}$ for the autonomous system (black circles) with error-limits presented by envelops of dotted-curves. (b) Average synchronization error across the rRNN for the driven ($\alpha = 0.8$) and autonomous system ($\alpha = 0$) in blue stars and black circles, respectively. Data was averaged over 100 different realizations of the rRNN.

smaller than amplitude dynamics of the autonomous rRNN, which according to Eq. (3.1) are limited to an amplitude of μ . This is particularly astonishing when keeping in mind that the rRNN approximates a chaotic attractor, and consequently the emulated self-driven system created from Eqs. (3.1) and (3.2) is highly sensitive to noise. This raises the question as to how the network can mitigate internal dynamics so efficiently.

The answer lies within the learning process. We demonstrate this by creating the rRNN's output y_{n+1}^{out} for $\alpha = 0$, however using the W^{out} previously learned for approximating y_{n+1}^T . From the resulting signal we discard the first 10 data points to avoid possible transient behavior. From the remaining 25 samples we

calculate the statistical amplitude variation via the signal's standard deviation σ according to

$$\sigma_{\alpha=0} = \left[\frac{\sigma(y_{n+1}^{out})}{\sigma(y_{n+1}^T)} \right]^2. \quad (3.4)$$

Measure $\sigma_{\alpha=0}$ evaluates the output weight's performance for suppressing autonomous dynamics. This corresponds to the crosstalk of autonomous rRNN dynamics into the system's readout. The black data in Fig 3.4(a) demonstrates how intricate both measures are related. Within regions R_1 , R_2 and R_3 , learning efficiently separate autonomous dynamics from transients induced by the sensory input. As revealed by the low values of $\sigma_{\alpha=0}$, in R_1 , R_2 and R_3 the rRNN there can approximate y_{n+1}^T well because the impact of autonomous rRNN dynamics on y^{out} is strongly reduced. There, both are confined to specific and well separated time scales. As a consequence the impact of perturbations, be it spontaneous dynamics or noise, not present in the training data can strongly be reduced. This is of significance as sensory information about a target system is fundamentally contaminated by noise. Noise within the injected information can therefore cause deviations. However, due to the rRNN's capability to dissipate noise, the network limits dynamics to a well defined range. We argue that this is achieved by minimizing the dissipating mechanism at the frequencies of the training signal, while timescales not present in the training signal are strongly attenuated.

The consequence of these mechanisms is that spontaneous internal activity of each node has an impact on the collective evolution of the network. Ultimately nodes are individual oscillators which influence each other through the internal weights. As illustrated in the right column of Fig. 3.3, this collective architecture can cause different levels of organization, that should influence the propagation of information within the network. The structure of the rRNN can be quantified by a measure of phase synchronization between the nodes. The standard deviation of all node responses individually averaged over time is measured against the rRNN's mean-field dynamical state:

$$\delta_{n+1} = \frac{1}{\mu} \sqrt{\frac{1}{N} \sum_{i=1}^N (x_{n+1}^i)^2 - \left(\frac{1}{N} \sum_{i=1}^N x_{n+1}^i \right)^2}. \quad (3.5)$$

Equation (3.5) is a global measure of phase synchronization between nodes [107]. A normalization by μ allows to associate δ_{n+1} to a synchronization error in the phase of the nonlinear function in Eq (3.1). As demonstrated by Fig 3.4(b), the global rRNN synchronization error significantly decreases

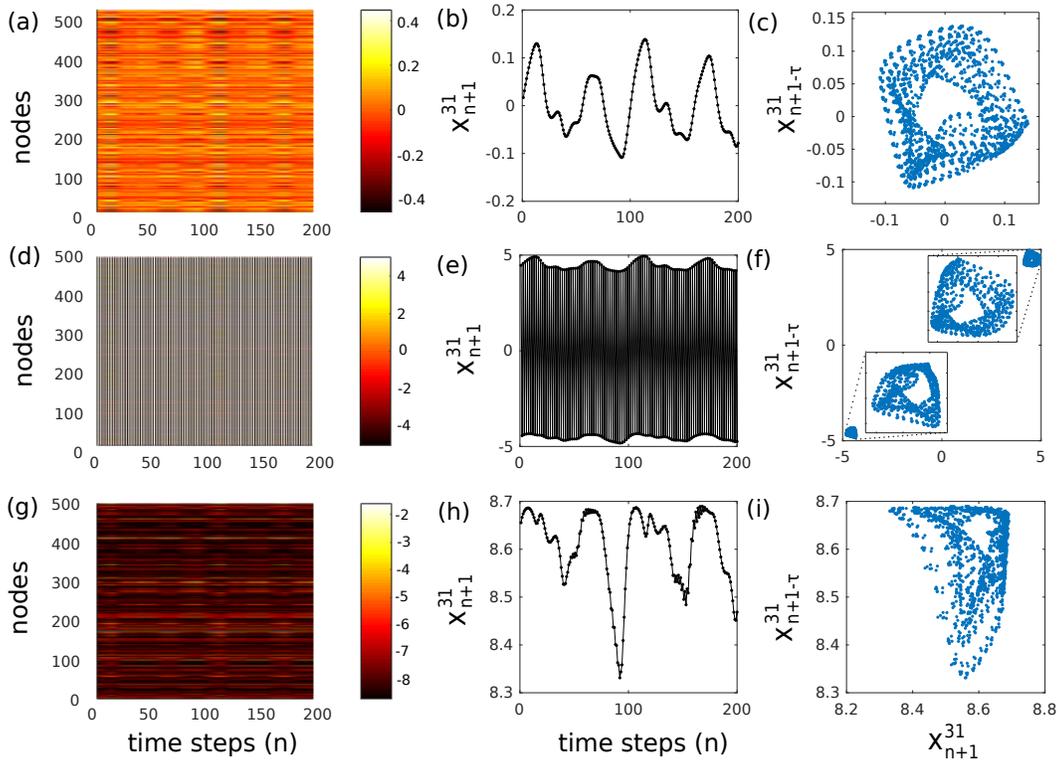


Figure 3.5: Spatio-temporal evolution of the rRNN (left column), time series recorded for node x_{n+1}^{31} (center column), and 2D projections of the reconstructed time series from node 31 (right column): (a-c) $\mu = 0.8$, (d-f) $\mu = 5$, and (g-i) $\mu = 8.7$. All dynamics obtained for the driven system ($\alpha = 0.8$).

in regions R_1 , R_2 and R_3 . Hence, synchronization and regular autonomous dynamics share the same parameter range. Region R_1 can clearly be separated into two sections. For $\mu \leq 1$ nodes are not well synchronized; the system is operating in the linear section of the nonlinear function. For $\mu > 1$ the operating point moves towards the nonlinear function's maxima and nodes start evolving in synchrony. Region R_3 is significantly more sensitive to μ when compared to R_1 and R_2 . This narrow range is due to the proximity to parameters resulting in chaotic dynamics. Such a sensitive operating point is less recommendable for example prediction using a noisy hardware or biological rRNN. In fact, in R_3 we find that the nodes are in steady states, however not as well synchronized as in R_1 , and R_2 .

The comparison between panels (a) and (b) of Fig 3.4 highlights the importance of the phase synchronization capability in autonomous regime, for good

prediction performance. Periodic dynamics ensures the discussed separation between timescales, while synchronization minimizes the resources learning has to dedicate for their suppression. For fully synchronized rRNN periodic dynamics it would already suffice to split W^{out} into symmetric groups of positive and negative readout weights, simply relying on statistical averaging between the two populations to suppress the autonomous dynamics' influence on the systems output state. The distribution of W^{out} within both groups can then be dedicated to the prediction of y_{n+1}^T . A reduced synchronization will consequently increase the resources learning has to assign for suppressing autonomous dynamics, leaving less freedom for optimizing prediction performance. Data shown in Fig 3.4 shows the statistical average obtained from 100 independent realizations of the system, each time defining a new random internal weight matrix W .

An extended qualitative analysis of the dynamics associated with good prediction performance is shown by Fig 3.5. For $\mu = 0.8$, Fig. 3.5(a) shows the spatio-temporal plot of all nodes when the input is injected in steady state regime R_1 . Column-shaped patterns throughout the entire spatio-temporal plot confirms phase synchronization between nodes. Indeed, these spatio-temporal patterns match with the injected signal. One example of how input information is preserved within the rRNN is revealed by Fig. 3.5(b), where the randomly chosen node 31 (x_{n+1}^{31}) shows a nonlinearly transformed version of the input. For a geometrical illustration of the information carried by the node, we illustrate the system's dynamic by reconstructing the MG attractor through Takens embedding Theorem [21]. From this embedding scheme, we obtain parameters of delay $\tau_0 = 12$ and dimensions $M = 4$ to fully reconstruct the MG state space. The state space of node x^{34} is in this example reconstructed based on the Takens theorem. A 2D projection of the state space is shown by Fig 3.5(c), which has strong similarities with the structure of MG chaotic attractor [112].

At $\mu = 5$ (regime R_2) we find the previously described periodic oscillations. The spatio-temporal plot shows once again a constant phase relation across all nodes, see Fig 3.5(d). However, it is not possible to identify any clear spatio-temporal features corresponding to the input timetrace. A closer look into the individual node evolution shows that the internal dynamics are still present, see Fig 3.5(e). We find that node dynamics consist of two contributions. Large amplitude oscillations at fast timescales correspond to the autonomous dynamics, while nonlinear transients induced by the input information are encoded in the slowly varying envelope. In Fig 3.5(f), the state space provides information about the geometrical structure of the modulation, where two different

2D projections of chaotic attractors are located at the extrema of the periodic oscillations.

Upon increasing the bifurcation parameter to $\mu = 8.7$ (regime R_3), the collective dynamics shown by Fig 3.5(g) resembles the spatio-temporal pattern from panel (a), even showing high phase synchronization. In this case, the node's fast periodic oscillation is substituted by irregularly appearing, noise-like epochs, see Fig 3.5(h). As in R_1 the fixed point of R_3 is a quasi-steady state, yet due to the narrow width of R_3 the rRNN is forced outside this stability window by even smaller fluctuations which in turn induce the noise like epochs. Figure 3.5(i) shows the effect of the noise on the reconstructed attractor. The noise strongly distorts the node responses away from the MG attractor. As induced by noise-like epochs, these distortions strongly hamper the determinism in the rRNN's response to the injected information. As reported in Chapter 2, the impact on spontaneous dynamics onto prediction has to be interpreted within the framework of the original attractor. It then becomes clear that the processed information might preserve some of its original features.

3.3 Preservation of information features in destabilized rRNNs

As illustrated by the different attractors of Fig. 3.5, spontaneous dynamics in the rRNN can cause distortions on the input dataset. These distortions hamper the neural network's ability to preserve essential features of original information. It is important to recall that this input sequence is chaotic, and by no means random. It is the result of complex, yet causal deterministic processes which define its complex temporal evolution. Then, predicting such a signal demands these causal relationships to be preserved within the neural network's dynamical state, providing a functional relationship to currently and previously injected information. For low error prediction it is therefore an essential condition that the network can serve as carrier and short term storage of injected information. Phase synchronization is therefore not sufficient to estimate if a rRNN complies with this condition. In order to quantify the information content preserved within the rRNN when stimulated by an input, we calculate the mutual information (MI) between the rRNN and the input signal for one realisation of W . This provides an estimation of how well the network is able to maintain the input information content [41], and hence is capable to capitalize from its own internal causal relationships for computation.

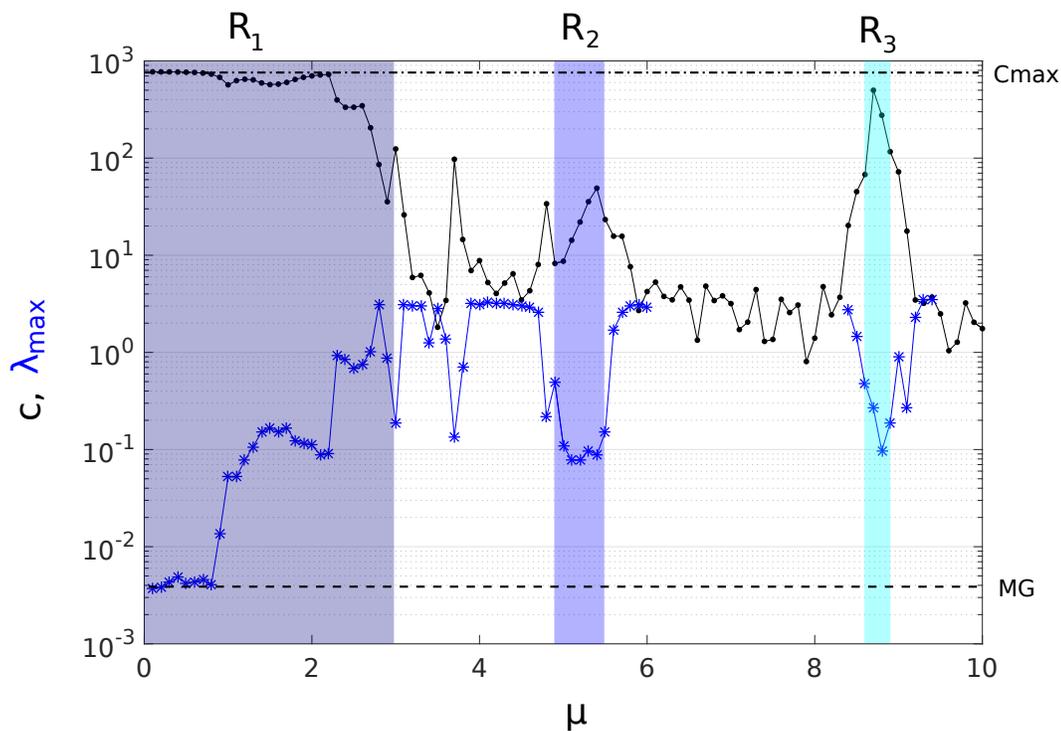


Figure 3.6: Memory capacity C (black dots), and maximal Lyapunov exponent λ_{max} (blue stars) of the rRNN, as functions of μ . Sections without λ_{max} values exhibit hyper-chaotic dynamics with more than one positive Lyapunov exponent. The dashed/dotted line indicates $\lambda_{max}^{MG}/C_{max}$, respectively

In this case, mutual information quantifies the amount of information shared between node responses x_{n+1}^i and input signal y_{n+1}^T [28]

$$MI_i = \sum_{x_{n+1}^i, y_{n+1}^T} \Psi(x_{n+1}^i, y_{n+1}^T) \log \frac{\Psi(x_{n+1}^i, y_{n+1}^T)}{\Psi(x_{n+1}^i) \Psi(y_{n+1}^T)}. \quad (3.6)$$

Mutual information MI therefore depends on the joint probability density function of x_{n+1}^i and y_{n+1}^T , $\Psi(x_{n+1}^i, y_{n+1}^T)$, as well as $\Psi(x_{n+1}^i)$, $\Psi(y_{n+1}^T)$ which are the probability density functions of x^i and y^T , respectively. If $MI_i \gg 0$ for the i th node, it preserves most dynamical properties of input y_{n+1}^T . Under these conditions the rRNN as a whole is capable to preserve the input information without significant loss of information, hence learning should be possible in principle. By accumulating all MI_i for the entire network (over every node) through the global measure $C_m = \sum_i (MI_i)$, we evaluate the capability of

the network to preserve and store significant features of the input in spite of the presence of network's spontaneous dynamics. Consequently, we define the measure C_m as the rRNN's memory capacity [108]. This index is calculated as a function of the bifurcation parameter and the results are shown in the black dotted trace in Fig 3.6.

In addition, we go beyond pure information content measures. Complex dynamical systems are typically classified using the rate of exponential divergence between neighbor trajectories, corresponding to their Lyapunov exponent, see Sec. 1.3.2. In particular, chaotic systems have a positive maximal Lyapunov exponent λ_{max} [25, 26, 27]. As the rRNN attempts to create a self-consistent representation of the chaotic target signal, comparing the network's λ_{max} to the target signal's maximal Lyapunov exponent provides a direct evaluation criteria. This index is calculated for each node i , λ_{max}^i , then the maximal Lyapunov exponent of the rRNN is $\lambda_{max} = \max(\lambda_{max}^i)$. In Fig 3.6 we show the rRNN's λ_{max} as blue data. Sections without λ_{max} values exhibit hyper-chaotic dynamics with more than one positive Lyapunov exponent.

As during our previous analysis, we find that regimes R_1 , R_2 , and R_3 have a strong impact upon these measures. Low-dimensional dynamics cause limited distortions to the rRNN's temporal evolution induced by the input signal. Consequently (i) memory capacity C_m is highest and (ii) the rate of divergence of rRNN trajectories is smallest. The rRNN is therefore capable to accurately preserve previous input information. In regime R_1 and R_3 steady states can be found, therefore the memory capacity C_m is higher than in R_2 , where the rRNN's spontaneous behavior is periodic in general. Our Lyapunov component analysis reveals that λ_{max} is kept small inside R_1 , R_2 , and R_3 due to their non-chaotic spontaneous features. In fact, for $\mu \leq 1$ in R_1 , λ_{max} of the network approaches with the one estimated for the input signal ($\lambda_{max}^{MG} \sim 3.6 \times 10^{-3}$). At this parameter, the rRNN is operated predominantly in its linear regime. Nevertheless, the fully nonlinear system ($\mu > 1$) combines oscillatory, spontaneous rRNN's dynamics with the injected input information. As the internal dynamics of the rRNN begin to exert influence over dynamics induced by the MG input, λ_{max} starts increasing accordingly. This behavior agrees well with the decrease of memory capacity, where the internal rRNN's dynamics will modify the probability distribution of the nodes. This demonstrates a strong correlation between the decline of spatio-temporal synchronization and the reduction in the system's memory capacity to approximate the deterministic, functional relationship of the prediction task.

3.4 Conclusion

Unlike neural networks run on a data-center, the human brain is not a special purpose computing machine, but parameters will most likely be optimized according to a compromise between partially competing demands. We therefore demonstrated that information preservation and synchronization inside a random network allow good prediction performance at parameters where learning in biological neural networks benefits. Based on a rRNN with a periodic nonlinear function we compare various regions of regular dynamics and highlight their importance of spatial synchronization upon prediction performance, mutual information and the stability of the neural network. Synchronization between nodes plays an essential role, but it is not sufficient to understand how information processing is successful in a rRNN beyond its fixed point. On the contrary, when linear regression is used to realize supervised learning, a causal relation between processed information and target is required.

We describe a rRNN predicting the future time-steps of a chaotic trajectory. Our results illustrate the importance of information flow, divergence and the suppression of signal components not present in the training data set. The rRNN's damped autonomous deviation $\sigma_{\alpha=0}$, mutual information MI and maximal Lyapunov exponent can be seen as complexity indicators for interpreting neural networks based on dynamical systems. Other than for the oscillatory state, chaotic responses were not capable to maintain important features of the input dynamic, resulting in a low prediction performance.

Among others, future work could investigate variations of connectivity network size. Their impact on synchronization and complexity indices might not be trivial, and what is their relationship with the computational capabilities of the rRNNs. Also, the application of mean-field theory on our network should be done in order to understand phase transitions as function of the bifurcation parameter μ .

Chapter 4

Electro-optic oscillator as delay neural network

Experimental realizations of delay differential equations (DDEs) implemented in Electro-Optic (EO) systems [113, 114, 115, 116] allowed the development of high performance broadband chaotic communications [117, 118], ultra-stable microwave sources [119, 120, 121], and random number generation [122]. The fundamental properties of such oscillators have also received significant attention [123, 124, 125, 126, 126]. Another important application of DDEs also emerged recently related to computer sciences applications. Here, the Ikeda delay equation (introduced in Sec. 1.2.3) is used as an integral component to design neuromorphic processors, from where it is possible to perform information processing and computing tasks [7, 127, 128]. As described in Sec. 1.2.3, one of the components used to describe the Ikeda system consists in a lowpass filter. In our current work we replace it for a highpass filter, resulting in the addition of an integral term to the original Ikeda equation, Eq. (1.20). As a consequence, the system is capable to show complex multiple time-scale dynamics, that has been investigated using several approaches like stability and bifurcation analysis [115, 129, 130, 131, 132]. More recently, it was shown that such hybrid dynamics can be understood through the paradigm of Liénard systems with attractive-repulsive branches [133]. However, the description of the waveform evolution requires a more extensive analysis of the structural properties of the entire system.

It has been also demonstrated that the structural shape of the state space depends largely on the nonlinear function's characteristics. In fact, some well established chaotic systems like Logistic, Tent and Bernoulli maps, Chua's circuit [134], Duffing-Holmes attractor [135] and the Ikeda equation are based on different kind of nonlinear functions. Their contribution in the generation of complex dynamics can be qualitatively described through the stretching and folding mechanism [136]. Stretching and folding results in exponential divergence of close orbits in certain state space regions. Trying to identify the relationship between nonlinear functions and their resulting dynamics, some reports show how complexity can be controlled by the nonlinear function extrema [137, 138, 139].

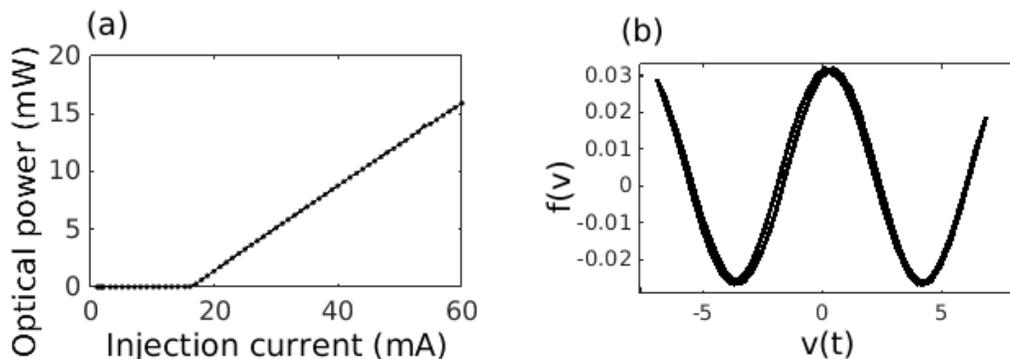


Figure 4.1: (a) Power-Injection curve of the laser, and (b) quartic function of the MZM, with a half-wave radio-frequency voltage $V_{\pi RF} = 3.8V$.

In the first part of this chapter, we propose a framework to characterize an autonomous delayed-feedback nonlinear optoelectronic oscillator based on the analysis of its nonlinear function. We analytically explain the integral term's impact for the modified Ikeda system. In particular, we characterize the route to spiral-like chaotic patterns in state space, as a function of feedback strength and operating point. In the second part, we utilize this setup in a steady state as an intrinsic component of a predictor for chaotic time series. This delay setup is used in both versions: lowpass and bandpass. The motivation to build predictors with two different filter structures is to characterize which of the Ikeda systems is more suitable for information processing. We investigate the dynamical properties of two different predictors, output-feedback and direct, performing short and long-term prediction. Finally, we implement a direct predictor in an electro-optic circuit, and compare the prediction performance with numerical simulations.

4.1 Autonomous electro-optical system

4.1.1 Electro-optic oscillator setup and model

The electro-optical system is composed by the following devices: a semiconductor telecom laser, used as a source of input light to the Mach-Zehnder modulator (MZM). The laser diode has a threshold current of $I_0 = 17.91$ mA, Fig. 4.1(a), an optical emission power of around 10 – 20 mW, and emission wavelength of approximately 1550 nm.

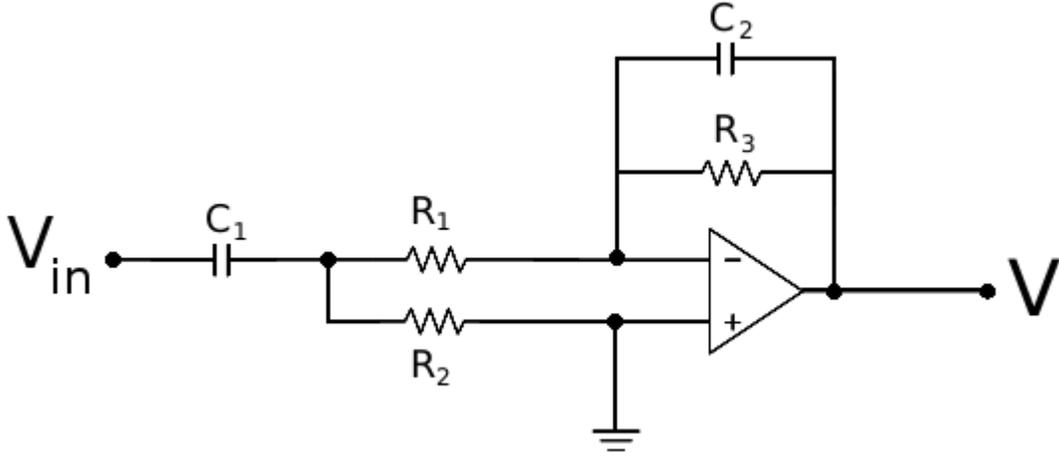


Figure 4.2: Schematic diagram of the electronic circuit, where $R_1 = 1.2K\Omega$, $R_2 = 150\Omega$, $R_3 = 1.2K\Omega$, $C_1 = 47\mu F$, and $C_2 = 222pF$.

A Mach-Zehnder Modulator (MZM) is connected to the laser by an optical fiber. The MZM performs the non-linear transformation through its modulation transfer function,

$$P_0(t) = P_{in} \cos^2 \left(\frac{\pi V(t)}{2 V_{\pi RF}} + \phi_0 \right), \quad (4.1)$$

where $V(t)$ is the input voltage applied to the radio-frequency (RF)-port, $V_{\pi RF} = 3.80$ V is the RF half-wave voltage, and $\phi_0 = \pi V_{DC}/2V_{\pi DC}$, $V_{\pi DC} = 6.64$ V. The MZM used for this experiment is an *AZ-1K1-12-PFA-SFA* electro-optic MZM from EOspace, with a non-linear transmission function that can span until a polynomial of degree 4, as we can see in the Fig. 4.1(b). The time delay is performed by a 4 km optical fiber, $\tau_D \simeq 19.472 \mu s$. A semiconductor photodiode detects the light, generating a current proportional to the intensity of the light absorbed,

$$V_{pd} = SP_{in} \cos^2 \left[\frac{\pi V(t - \tau_D)}{2 V_{\pi RF}} + \phi_0 \right], \quad (4.2)$$

with S as the sensitivity of the photodiode, $S = 0.9$ mA/mW. (5) The band-pass filter has cutoff frequencies $f_l = 22.5$ Hz and $f_h = 0.6$ MHz, according to an electrical scheme presented in the Fig. 4.2. The integro-differential equation describing the filter is

$$V_{in}(t) = \left(1 + \frac{f_l}{f_h} \right) V(t) + \frac{1}{2\pi f_h} \frac{dV(t)}{dt} + 2\pi f_l \int_{t_0}^t V(t') dt', \quad (4.3)$$

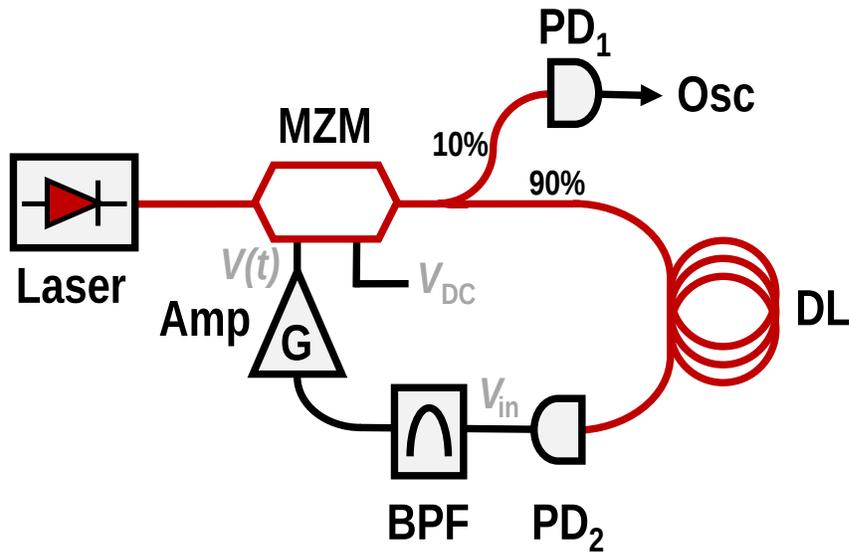


Figure 4.3: Schematic diagram of the nonlinear time-delayed EO setup. MZM: Mach-Zehnder Modulator; PD: Photodiode; DL: Delay-Line; BPF: Bandpass Filter; Amp: RF amplifier.

where $2\pi f_l = 1/R_2C_1$ and $2\pi f_h = 1/R_1C_2$.

The equation of the full nonlinear feedback system is obtained relating the input of the filter with the output of the photodiode $V_{in} = GV_{pd}$, with gain $G = 3.3$, that is combining Eqs. (4.2) and (4.3):

$$\left(1 + \frac{f_l}{f_h}\right) V(t) + \frac{1}{2\pi f_h} \frac{dV(t)}{dt} + 2\pi f_l \int_{t_0}^t V(t') dt' = GSP_{in} \cos^2 \left[\frac{\pi}{2} \frac{V(t - \tau_D)}{V_{\pi RRF}} + \phi_0 \right]. \quad (4.4)$$

Assuming that the bandpass filter results from the cascade of first-order lowpass and high-pass filters with $f_l \ll f_h$, it can be shown that system dynamics are described by a normalized integro-differential delay equation, which is explicitly written as:

$$x(t) + \tau \frac{dx}{dt}(t) + \frac{1}{\theta} \int_{t_0}^t x(\xi) d\xi = \beta \cos^2[x(t - \tau_D) + \phi_0], \quad (4.5)$$

where,

$$x(t) = \frac{\pi V(t)}{2V_{\pi RF}}; \phi_0 = \frac{\pi V_{DC}}{2V_{\pi DC}}; \beta = \frac{\pi GSP_{in}}{2V_{\pi RF}}; \quad (4.6)$$

$$\tau = \frac{1}{2\pi f_h}; \theta = \frac{1}{2\pi f_l}. \quad (4.7)$$

For the dynamical system, the main control parameter β can be tuned through the laser power P_{in} . Furthermore, ϕ_0 stands for the bias offset phase of the MZM. Both, β and ϕ_0 , are bifurcation parameters for the oscillator. τ and θ are the characteristics fast and slow response times of the filter.

For mathematical purposes, it is convenient to rewrite Eq. (4.5) as a set of two dimensionless coupled first-order DDEs with respect to the dimensionless time $\zeta = t/\tau_D$, resulting in

$$\begin{aligned} \varepsilon \dot{x}(\zeta) &= -x(\zeta) - \delta y(\zeta) + \beta \cos^2[x(\zeta - 1) + \phi_0], \\ \dot{y}(\zeta) &= x(\zeta); \end{aligned} \quad (4.8)$$

with $\varepsilon = \tau/\tau_D$, and $\delta = \tau_D/\theta$.

4.1.2 Dynamics of the autonomous system

In the following, the dynamical characteristics of the time-delayed EO oscillator are briefly analysed. A first step is the stability analysis of the trivial fixed point, which can be investigated by deriving an eigenvalue characteristic equation obtained after assuming perturbations proportional to $e^{\lambda\zeta}$ in Eq. (4.8):

$$\lambda^2 + \lambda[1 + \beta \sin(2\phi_0)e^{-\lambda}] + \varepsilon = 0. \quad (4.9)$$

The analysis of this transcendental equation reveals that the trivial fixed point is stable when $|\beta \sin(2\phi_0)| < 1$; beyond this limit, a Hopf-bifurcation might arise. In this case, the solution of Eq. (4.9) becomes purely imaginary ($\lambda = i\omega$), and we obtain the following equations:

$$\begin{aligned} \omega \tan \omega &= \delta - \varepsilon\omega^2, \\ -\beta \sin(2\phi_0) &= (\cos \omega)^{-1}. \end{aligned} \quad (4.10)$$

The first equation of Eqs. (4.10) defines the eigenmodes of the system, allowing us to analytically calculate the Hopf angular frequency. The second equation describes the section of the nonlinear function where the oscillation is possible, as a function of each angular frequency. For sections of the nonlinear function

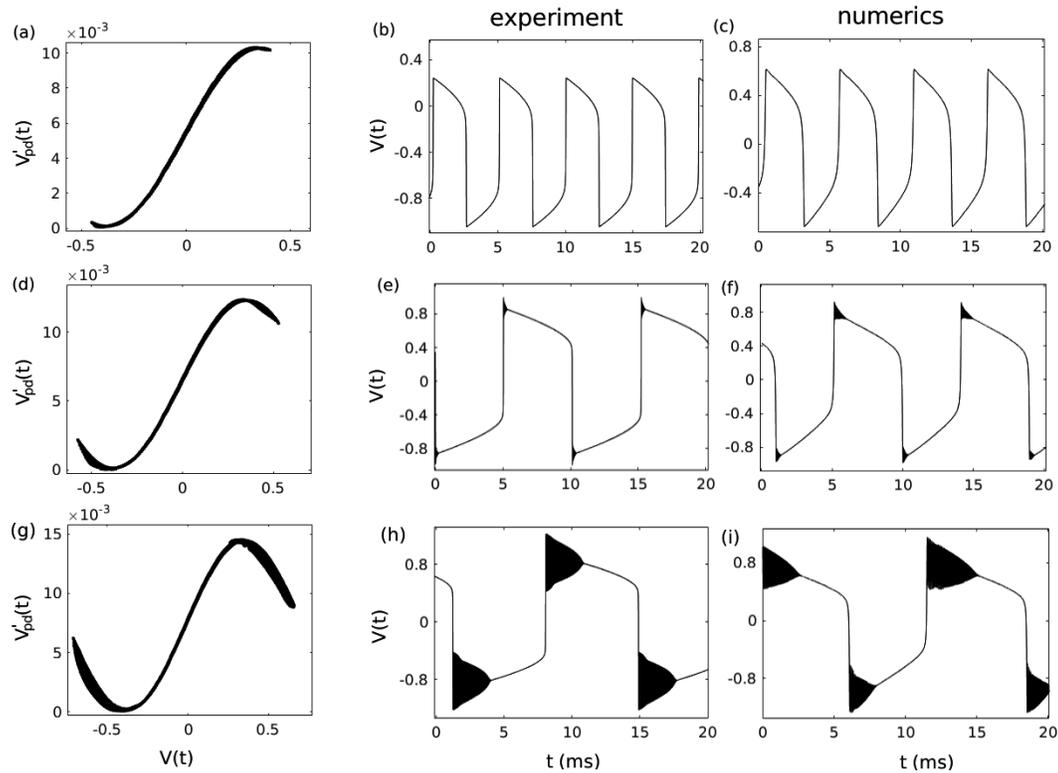


Figure 4.4: Experimental and numerical results normalized gain β set to 1.4 (first row), 1.7 (second row), and 2.2 (third row). Left column: Transmission function of the MZM when operated at the middle inflection point with positive slope ($\phi_0 = -\pi/4$). Central column: experimental timetraces; Right column: numerical timetraces.

with a positive slope, we find that the eigen angular frequency is $\omega \simeq \sqrt{\delta}$, corresponding to a slow oscillation. Panels (a,b,c) of Fig. 4.4 show corresponding slow oscillations. For operation along the negative slope we obtain $\omega \simeq \pi$, hence the period of oscillation is approximately twice the delay time.

When $-\beta \sin(2\phi_0) > 1$, a mixed-mode oscillation appears [133], see panels (d,e,f) of Fig. 4.4. It results from the superposition of two waveforms: (i) a slow-scale periodic signal, with 5 ms period which is consistent in magnitude with the value obtained at the Hopf threshold for the slow eigenfrequency, $\omega = \sqrt{\delta}$, leading to a physical period of $2\pi\tau_D/\omega = \sqrt{2\pi\tau_D/f_1} \simeq 2.3$ ms. There, the curves at the top and bottom of the time series correspond to relaxation oscillations separated by sudden jumps. (ii) The fast-scale dynamics

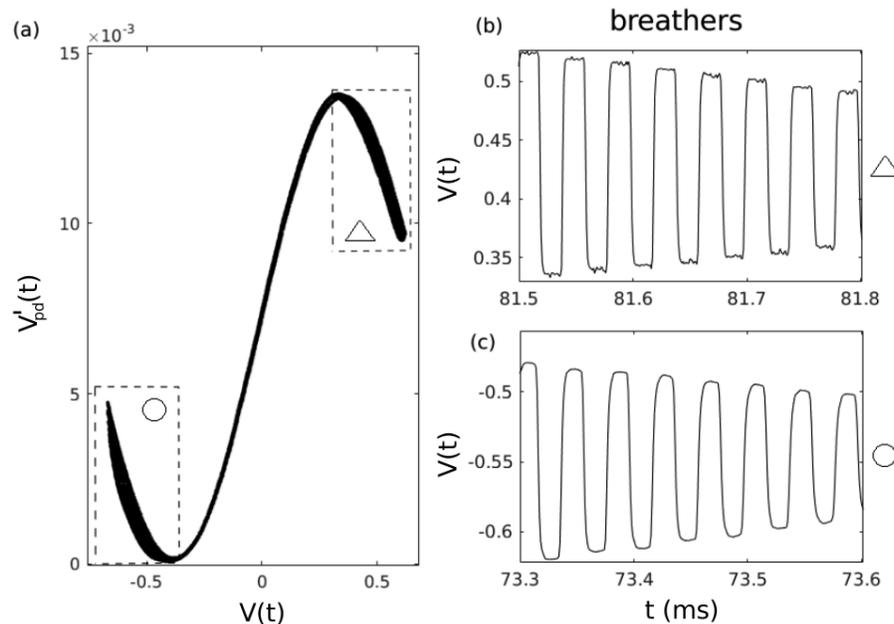


Figure 4.5: (a) MZM transmission function with labels $\{\triangle, \circ\}$ for the portions with negative slope. Zoom at the top (b) and bottom (c) of the time series showing mixed-mode dynamics (see Fig. 4.4(i)), where $\beta = 2.2$ and $\phi_0 \simeq -\pi/4$. Both with period $T_P = 40\mu s = 2\tau_D$.

generated in the extrema's neighbourhood of the periodic time series. Finally, dynamics with a strong coexistence of both timescales are shown in Fig. 4.4(g,h,i). Through a qualitative inspection of the nonlinear function in Fig. 4.4(a,d,g), one can observe that the sections of the nonlinear function with a negative slope increase with β . This causes the emergence of fast-scale oscillations with period $T_P = 40\mu s = 2\tau_D$ at the extrema of the slow envelop, see Fig. 4.4(e,f,h,i). Here, the decay in the temporal envelopes for the fast-scale oscillations are related to the existence of relaxation epochs. Such epochs are fast time scale oscillations occurring in the neighborhood of the nonlinear function extrema and along its negative slopes. This highlights the importance of the nonlinear function's shape in the actual dynamical solution. In addition, one can see that the period of the slow envelop continuously increases with growing β , from 2.3 ms at the slow envelope Hopf threshold ($\beta \simeq 1$), to 14 ms in Fig. 4.4(g,h,i), through 5 ms in Fig. 4.4(a,b,c) and 10 ms in Fig. 4.4(d,e,f). Such a period growth is consistent with the analytical study reported in [132].

By zooming in the extrema's neighbourhood of the time series when $\{\beta =$

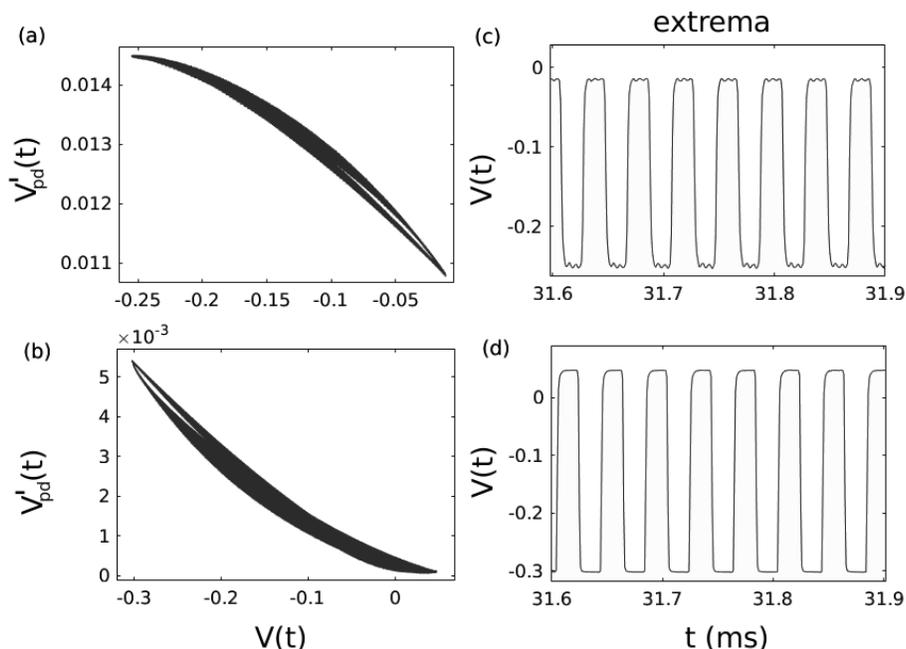


Figure 4.6: MZM transmission functions for $\beta = 2.2$, if $\phi_0 \simeq 0$ (a) and $\phi_0 \simeq -\pi/2$ (b). Time series associated to the nonlinear function with $\phi_0 \simeq 0$ (c) and $\phi_0 \simeq -\pi/2$ (d). Both with period $T_P = 40\mu s = 2\tau_D$.

2.2, $\phi_0 = -\pi/4$ }, it can be seen that fast-scale oscillations have an obvious correspondence with the dynamics generated by the isolated parts of the nonlinear function with negative slope, where $\{\beta = 2.2, \phi_0 = -\pi/2; 0\}$. Figures 4.5 and 4.6 show time series for $\phi_0 = -\pi/4$ and $\phi_0 = \{-\pi/2; 0\}$, respectively. In Fig. 4.5, panels (b) and (c) focus on the top and bottom of the nonlinear function. In Fig. 4.6, these two extrema are disconnected, with dynamics shown in panels (c) and (d). Under both conditions, dynamics are qualitatively comparable, having the same oscillation period $T_P \simeq 2\tau_D \simeq 40 \mu s$ as predicted by Eqs. (4.10) for the first fast eigenmode, with a very similar theoretical value compared to standard Ikeda model with double delay periodicity.

4.1.3 The Liénard-Ikeda approach

In this section, we analyse the interaction between the two time-scales involved in the regime of breather oscillations. We therefore focus on the configuration where $\phi_0 = -\pi/4$, resulting in perfectly symmetric oscillations. One should

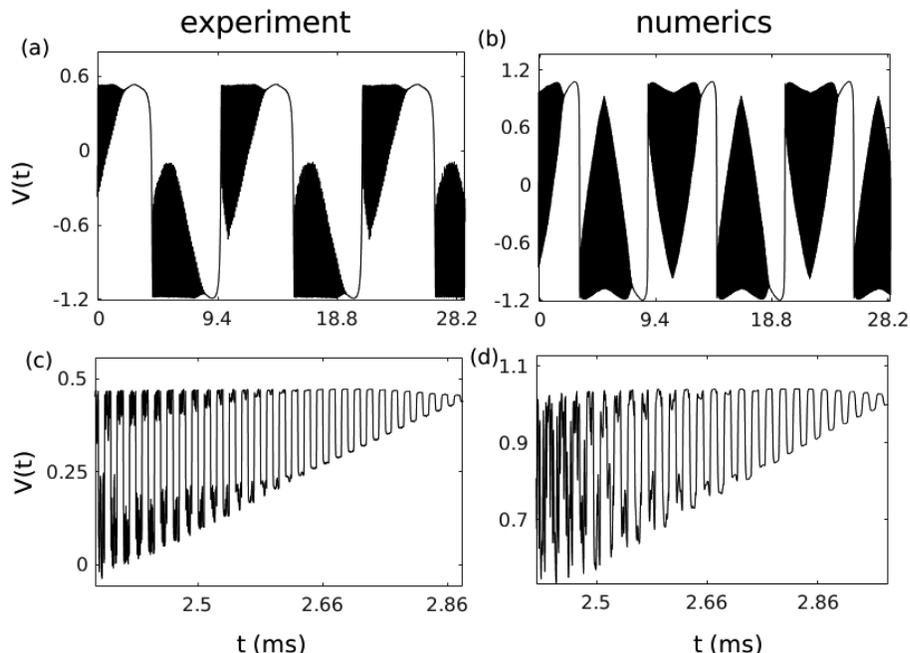


Figure 4.7: (a), (b) Breathers-like dynamics can be generated by changing periodically the MZM operating point in the Ikeda EO setup, and (c), (d) zoom at the top of the time series.

notice that the particular case $\Phi_0 = -\pi/4$ is used here for analytical convenience only, resulting in the perfect symmetry conditions assumed in the original Liénard limit cycle theorem. Deviations away from $-\pi/4$ however do not modify qualitatively the observed waveform, essentially changing the duty cycle of the slow motion. This was already underlined in previous references such as [133]. In the particular case of $\Phi_0 = -\pi/4$, Eq. (4.8) can be rewritten as:

$$\ddot{x} + \dot{x} - \beta \cos[2x(\zeta - 1)]\dot{x}(\zeta - 1) + \varepsilon x = 0. \quad (4.11)$$

In order to recover the textbook equation for Liénard system [140] from the previous equation of a bandpass EO delay dynamics, one needs to apply a new change of variable $s = \zeta \sqrt{\delta/\varepsilon}$. Doing this and assuming that the delay can be neglected compared to the time scales of the Liénard cycle, one indeed obtains:

$$\ddot{x} + F'(x)\dot{x} + G'(x) = 0, \quad (4.12)$$

where

$$F'(x) = \frac{1}{\sqrt{\varepsilon\delta}} [1 - \beta \cos(2x)] \quad (4.13)$$

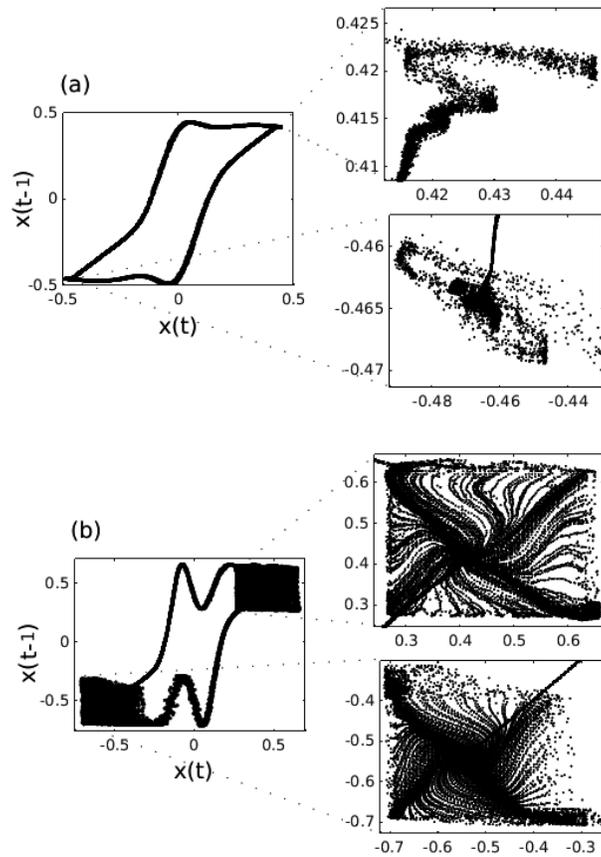


Figure 4.8: First-return maps from the time series: (a) $\beta = 1.7$ with the corresponding zoom at the top and bottom extrema, (b) $\beta = 2.2$ with the corresponding zoom at the top and bottom extrema, where $\phi_0 = -\pi/4$.

is an even function scaling the nonlinear damping, while $G'(x) = x$ is an odd function corresponding to the restoring force. This associated function $F(x)$ is determined from $F'(x)$ in Eq. (4.13). The Liénard plane is thus defined as:

$$\dot{x} = y - F(x), \quad (4.14)$$

$$\dot{y} = -g(x), \quad (4.15)$$

where

$$F(x) = \int_0^x F'(\xi) d\xi = \frac{1}{\sqrt{\varepsilon\delta}} \left[x - \frac{1}{2}\beta \sin(2x) \right], \quad (4.16)$$

and $g(x) = G'(x)$ are smooth odd functions. From here it is possible to obtain the Liénard xy -plane, which provides the exact solutions for the nonlinear

transformations shown in panels (a,d,g) of Fig. 4.4. According to the theorem ruling Liénard equation and its possible limit cycle solutions, Eqs.(4.14-4.16) have stable limit cycles if $F(x)$ has exactly three zeros with $F'(0) = (\varepsilon\delta)^{-\frac{1}{2}}[1 - \beta] < 0$ if $\beta > 1$, and $F(x) \rightarrow \infty$ if $x \rightarrow \infty$ [140]. A more detailed analysis of the Liénard derivation and bandpass Ikeda dynamics, as well as its detailed discussion in the xy -plane, can be found in [133].

The previous transformation of the bandpass EO delay dynamics into a Liénard system is helpful to discuss our main finding concerning the regime depicted in Figs. 4.4 to 4.8. In the EO bandpass delay system with moderate feedback ($\beta = 1$ to ca. 3), fast and slow timescales are in fact nonlinearly coupled, however in an unidirectional way. Fast dynamics obey the standard Ikeda equation, while slow dynamics follow a Liénard limit-cycle. The slow Liénard cycle actually drives the operating point Φ_0 , around which a fast conventional Ikeda dynamics (i.e. low pass model, without the integral term) develops. To demonstrate this physically, we have performed a simple experiment in which the Liénard compound is replaced by an external triangular waveform modulating the offset phase parameter. The triangular shape of the waveform is motivated by the fact that the external drive is intended to replace the integral variable y of the original bandpass delay dynamics. This variable is indeed the long time scale integral of the variable x which can be approximated, on average (due to the slow integration time of concern), by a constant, whether positive or negative. The corresponding integral is then simply a triangular waveform. Such a system is then modeled by:

$$\varepsilon \frac{dz}{dt}(t) = -z(t) + \beta \cos^2[z(t) + \Phi_0 + u(t)] \quad (4.17)$$

where $u(t)$ is a triangular waveform having an amplitude and a period tuned, so that its effects emulate the actually observed slow oscillation of the Liénard limit-cycle. One could notice that the previous Eq. (4.17) can be obtained from the bandpass Ikeda model in Eq. (4.8), setting $z(t) = x(t) + \delta y(t)$ and $u(t) = -\delta y(t)$, and assuming that the term $\varepsilon \delta \dot{y}$ is a negligible derivative in time compared to $\varepsilon \dot{z}$, due to the smallness of δ . Experimental and numerical results are shown in Fig. 4.7, where both, the global shape as well as finer details at fast time scales are in excellent agreement. Experimentally, the forcing $u(t)$ of the standard Ikeda system was implemented by modulating ϕ_0 through the MZM's DC input, with amplitude $V_{DC} = 5.9$ V and frequency 211 Hz. By using these values we approximate the slow periodic envelope of the bandpass Ikeda with the triangular signal, whose frequency is typically around 200 Hz. Furthermore, the amplitude of the triangular signal spans

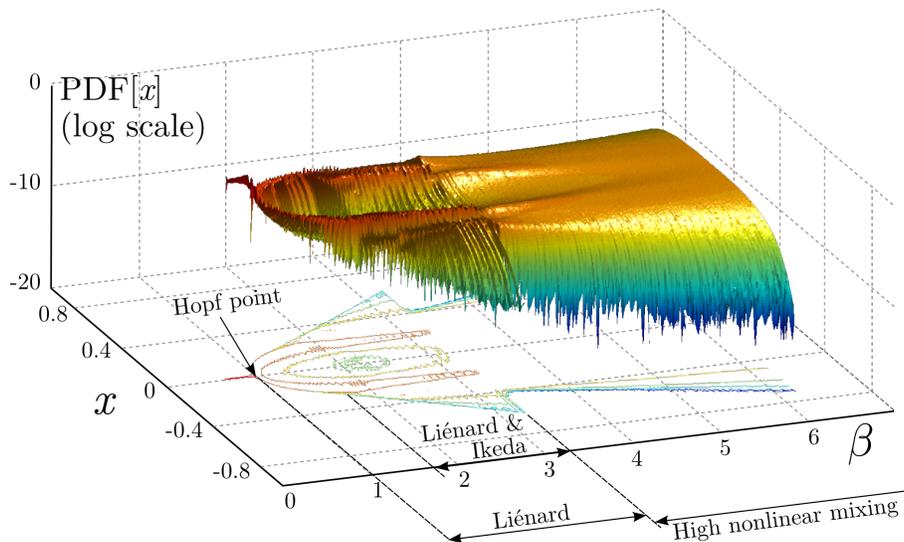


Figure 4.9: Numerical simulations of distributions of the orbits $x(t)$ for different feedback gain values β and $\phi_0 = -\pi/4$.

the full extrema neighbourhood of the nonlinear function, $\phi_0 \in \{-\pi/2, \dots, 0\}$, where the Ikeda-like dynamics are dominant. The timetraces exhibit Ikeda-like oscillations combined with periodic breathing at the frequency of the external modulation. Consequently, it can be deduced that the evolution of the system displays Ikeda-like oscillations close to the extrema neighbourhood, while the integral part induces slow timescale Liénard-like oscillations.

It is informative to illustrate the system's dynamics in a 2-dimensional phase space using an attractor reconstruction technique. This procedure is carried out by using an additional time delay coordinate $x(\zeta - 1)$, which allows to project the original attractor onto a 2-dimensional phase space. In order to implement this method, we have recorded an experimental timetrace from the EO experimental setup, and used it to obtain the coordinates for the time-delay based 2D-embedding. Figure 4.8 shows 2-dimensional projections of the reconstructed attractors in the phase space for two types of breathers, where $\beta \in \{1.7, 2.2\}$. The existence of a large limit cycle is revealed by the slow-scale dynamics corresponding to the Liénard dynamics. The spiral-like trajectories in the limit cycle represent the fast-scale Ikeda-like solutions in the 2D-projection. This behavior is maintained even when the fast-scale dynamics has a chaotic nature. The spiral-shaped pattern of the attractor is related to the relaxation dynamics of the system induced by the slow-scale periodic oscillation, which yields a decaying in time envelope for the Ikeda-like wave-

forms. This spiral pattern is therefore the topological signature of a global interaction of two different dynamics, which are interacting nonlinearly with different timescales, however unidirectionally from the slow motion (θ) into the fast ones (τ_D and τ).

As illustrated in Fig. 4.9, when β is further increased (ca. above 3.3), stronger time scales interplay results in the vanishing of the Liénard slow limit cycle and gives rise to fully developed chaos. We anticipate that such fully developed chaos consists of a more complex and probably bidirectional time scale nonlinear mixing, as clearly shown by the nicely smoothed probability density function profile after the vanishing of the Liénard envelope (see Fig. 7).

4.2 Electro-optic oscillator for information processing

We have studied the dynamical properties of the Ikeda system in the temporal domain. There, we investigated in detail the coupling between the fast-scale behavior associated to a characteristic lowpass Ikeda behavior, and slow-scale dynamics associated to a Liénard limit-cycle bandpass Ikeda, for $\phi_0 = -\pi/4$. From this study, we conclude that it is possible to approximate characteristic lowpass Ikeda behavior in the bandpass Ikeda.

In the following, we introduce a second study related to the capabilities of this system to process information. It starts by describing the spatio-temporal dynamical features of the Ikeda system, that can lead to a structural comparison with spatial networks that process information. From the spatio-temporal approach it is plausible to understand how many spatial variables we can find to be used as nodes. Here, we work with two different kind of Ikeda systems, which are differentiated by the use of lowpass or bandpass filters.

4.2.1 Spatio-temporal approach

Let us introduce the Ikeda systems represented as an infinite system of coupled ordinary differential equations (ODE) [17],

$$\begin{cases} \dot{z}_0(t) = f(t), \\ \dot{z}_1(t) = g(t), \\ \dot{z}_i(t) = \frac{n}{R}(z_{i-1}(t) - z_i(t)); \end{cases} \quad (4.18)$$

for $i = 2, 3, \dots, n \rightarrow \infty$. The bandpass Ikeda is described by $f(\iota) = -z_0(\iota) - \rho z_1(\iota) + \beta \cos^2(z_n(\iota) + \phi_0)$, and $g(\iota) = z_1(\iota)$. The lowpass Ikeda system by $z_1(\iota) = 0$, then $f(\iota) = -z_0(\iota) + \beta \cos^2(z_n(\iota) + \phi_0)$, and $g(\iota) = 0$. With τ as the fast response time of the filter, we normalize variables by τ , defining $\iota = t/\tau$ as the dimensionless time, $R = \tau_D/\tau$ the dimensionless time delay, $\rho = \tau/\theta$ the dissipative coefficient depending on θ (the slow response time of the filter), and the coupling term between spatial elements is n/R . This representation is fulfilled if and only if

$$z_n(\iota) \rightarrow x(\iota), \quad \text{when } n \rightarrow \infty, \quad (4.19)$$

with

$$\lim_{n \rightarrow \infty} R_n = R, \quad (4.20)$$

where, the time delay R has been represented as an interval of n elements. When these elements are infinite ($n \rightarrow \infty$) we can write the DDE as an infinite system of coupled ODEs. Each ODE models the temporal evolution of one dynamical variable $z_i(\iota)$, which therefore corresponds to the i th spatial variable. Consequently, we have established a transformation into a pseudo space-time domain ($i - \iota$) of the purely time representation of the Eqs. (4.8) and (1.21).

In order to solve Eq. (4.18), we use the Euler integration scheme. The advantage of following this procedure is that it allows us to use the space-time representation to get the solution of the DDE. Its discrete-time approximation is given by

$$\begin{cases} z_0(k+1) = z_0(k) + 3hf(k)/2, \\ z_1(k+1) = z_1(k) + 3hg(k)/2, \\ z_s(k+1) = z_{s-1}(k); \end{cases} \quad (4.21)$$

where $h = R/(\Upsilon + 1/2)$, and $2 \leq s \leq \Upsilon \rightarrow \infty$. The functions $\{f, g\}$ are defined for the bandpass Ikeda as $f(k) = -z_0(k) - \rho z_1(k) + \beta \cos^2(z_M(k) + \phi_0)$, $g(k) = z_0(k)$, and for the lowpass Ikeda as $f(k) = -z_0(k) + \beta \cos^2(z_M(k) + \phi_0)$, $g(k) = 0$. Here, space-time ($s - k$) is defined by spatial variable s , which indicates the number of spatial variables s needed to describe the entire system, and k describes the discrete-time evolution. In fact, spatial variable defines how many initial conditions over the range $-R < k < 0$ are required in our approximation of the continuous-time delay oscillator.

We perform the numerical simulation by setting $M = 210$, $R = 63$, $\rho = 3.67 \times 10^{-5}$, and $\phi_0 = -0.6\pi$. The spatio-temporal ($s - k$) patterns for $\beta = \{1.76, 2.3, 3\}$ in the lowpass Ikeda oscillator, and $\beta = \{2.6, 3.2, 6\}$ in the bandpass Ikeda oscillator are shown by Fig. 4.10(a,b,c) and Fig. 4.10(d,e,f),

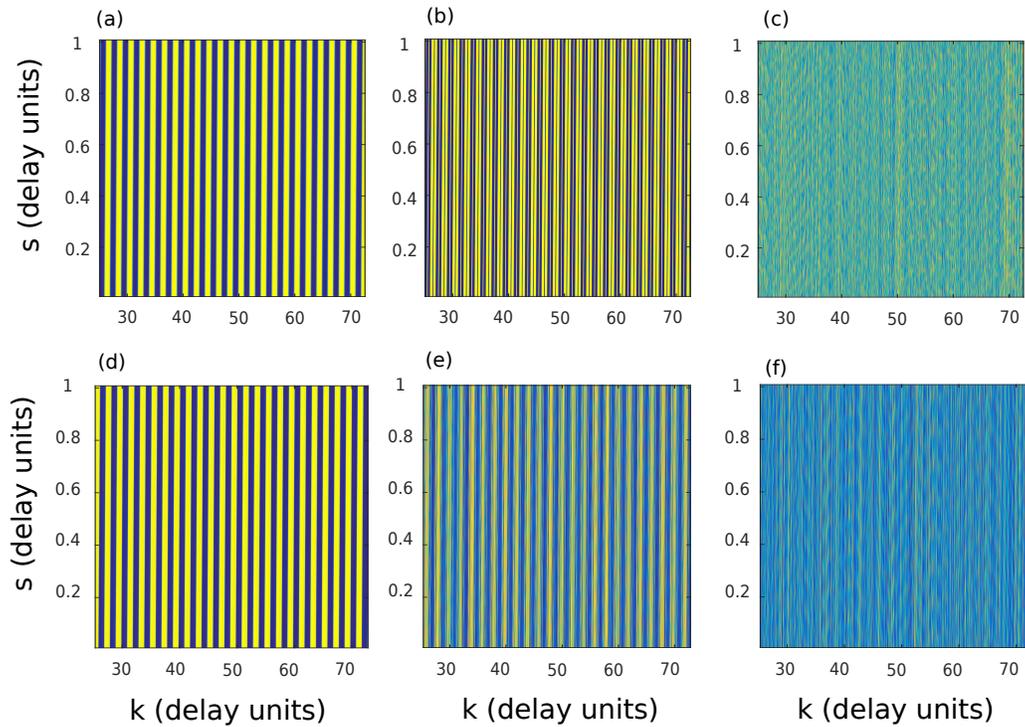


Figure 4.10: Spatio-temporal ($s - k$) patterns for $\beta = \{1.76, 2.3, 3\}$ in the lowpass Ikeda oscillator (a-c), and $\beta = \{2.6, 3.2, 6\}$ in the bandpass Ikeda oscillator (d-f).

respectively. Vertical axes range over one delay, while horizontal axes cover around 50 delay units. These patterns give insights about the hidden spatio-temporal dynamics evolving in delay oscillators. The left panel of Fig. 4.10 shows regular spatio-temporal structure of periodic solutions. This spatial regularity is approximately preserved even if the solutions are chaotic, see central panel of Fig. 4.10. Finally, the previous shown spatio-temporal patterns have been attenuated, corresponding to hyper-chaotic solutions, see right panel of Fig. 4.10. At this point, we highlight that after the transformation of the delay coupled oscillator into the coupled spatio-temporal system, the resulting dynamics share strong similarities with the spatial network introduced in Chapter 3. The element in common of both dynamical systems is the sinusoidal nonlinear function. In these cases, the sinusoidal function allows the existence of Hopf bifurcation as part of its route to chaos.

As it was shown, spatial representation in the Ikeda systems is identified through the variable s , where each component in their domain corresponds to

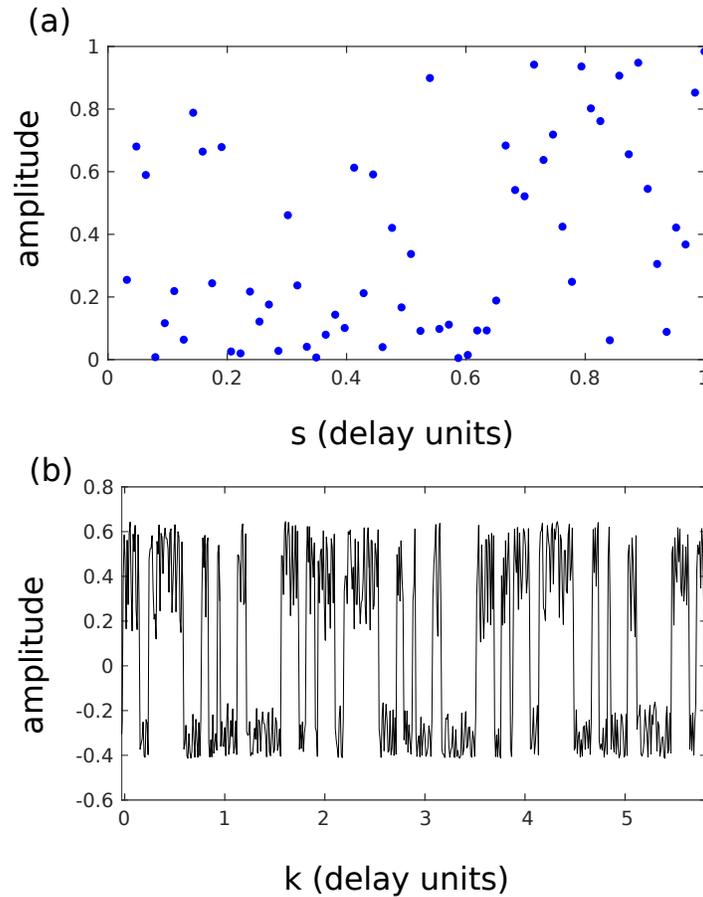


Figure 4.11: Distribution of the dynamical variables along the (a) s -spatial at $k = 1$ and (b) k -temporal at $s = 3$ variables in the lowpass Ikeda oscillator at $\beta = 2.44$.

a spatial coordinate. However, these spatial coordinates correspond to temporal terms defined differently. For example, in a single delay unit s we have R elements defined as coordinates, the separation in time between these coordinates is equal to h . Figure 4.11 shows how the spatial coordinates are distributed along the s -domain in the lowpass Ikeda oscillator. In Fig. 4.11(a), the positions in amplitude of the R spatial coordinates are shown for $\beta = 2.44$ and $k = 1$, where each of them is identified by a unique amplitude value. In Fig. 4.11(b) we show the chaotic temporal evolution of one of the coordinates at $s = 3$.

Until this point, our study evidences that we can obtain spatio-temporal rep-

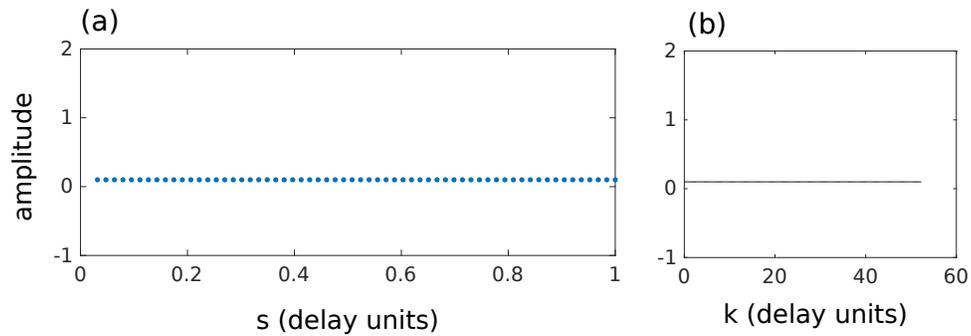


Figure 4.12: Distribution of the dynamical variables along the (a) s -spatial at $k = 1$ and (b) k -temporal at $s = 3$ variables in the lowpass Ikeda oscillator at $\beta < 1.8$.

representations of the Ikeda systems, where the dynamical variables are separable and distinguishable. In addition, the whole system of equations is found to be made of an infinite set of coupled ODEs. The coupling term is defined as discrete interactions between nodes, unlike the approach introduced in Sec. 1.4.7 where the coupling is defined via the linear impulse response of the delay oscillator. Hence, we can start pointing out the conditions to use our infinite system of coupled ODEs as artificial neural networks (ANNs). The network-like model defined here will be built by considering each spatial individual place in the space domain as a node.

As introduced in previous chapters, information processing is better performed at resting state of the network. In general, the resting state can be thought in terms of fixed points, i.e. it does not have time evolution until it is perturbed by an external stimulus. From our infinite system of coupled ODEs, we can also have all the dynamical variables in resting state. As an illustration of this fact, for low values of the gain feedback parameter, $\beta < 1.8$ in lowpass Ikeda, it is plausible to find individual fixed points. The R spatial coordinates at $k = 1$ are all organized along a constant amplitude, see Fig. 4.12(a). Figure 4.12(b) shows an example of the temporal domain of the spatial coordinate $s = 3$.

The full set of possible dynamical states are shown by the bifurcation diagrams varying β in Fig. 4.13 for the (a) bandpass and (b) lowpass Ikeda systems. These diagrams show that there are large intervals for the feedback gain $\beta = (0, 1.8)$ for the lowpass, and $\beta = (0, 2.7)$ for the bandpass Ikeda, where the fixed points are stable. The case of $\phi_0 = -\pi/4$ in the bandpass representation, studied in the first part of this chapter, shows that the fixed

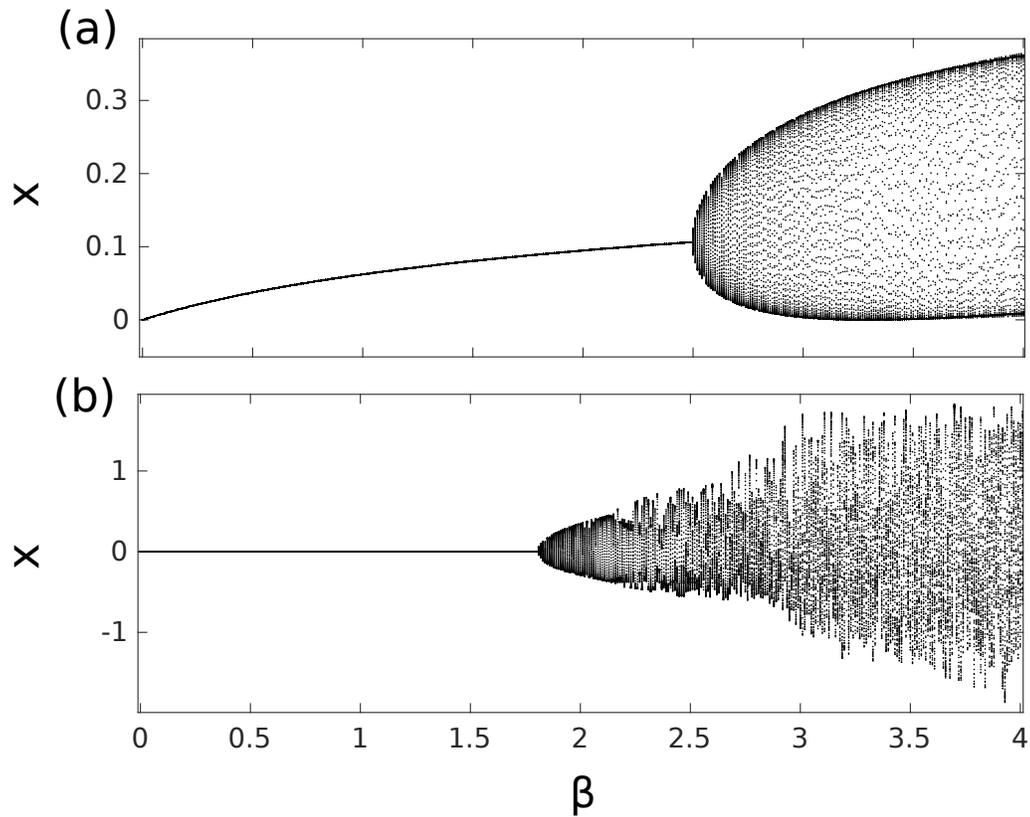


Figure 4.13: Bifurcation diagram for the (a) lowpass and (b) bandpass Ikeda oscillators at $\phi_0 = -0.6\pi$.

point is stable for a shorter interval $\beta = (0, 1)$. Otherwise, in the operating point at $\phi_0 = -0.6\pi$, we can expect an extensive stable manifold that can be further used for information processing.

4.2.2 Delay reservoir computing for prediction of chaotic time series

The scheme of our delay reservoir computing (DRC) [7, 8, 141, 142] concept for prediction is shown by Fig. 4.14. It illustrates the Eq. (4.21) with $f(k) = -z_0(k) - \rho z_1(k) + \beta \cos^2(z_M(k) + \alpha u_{in} + \phi_0)$ and $f(k) = -z_0(k) + \beta \cos^2(z_M(k) + \alpha u_{in} + \phi_0)$ for bandpass and lowpass Ikeda, respectively. Spacing between nodes in DRC are defined in this approach by $\delta\tau$. Thus, our network has R spatial variables, then the connectivity of our network is given by $R/\delta\tau$. The

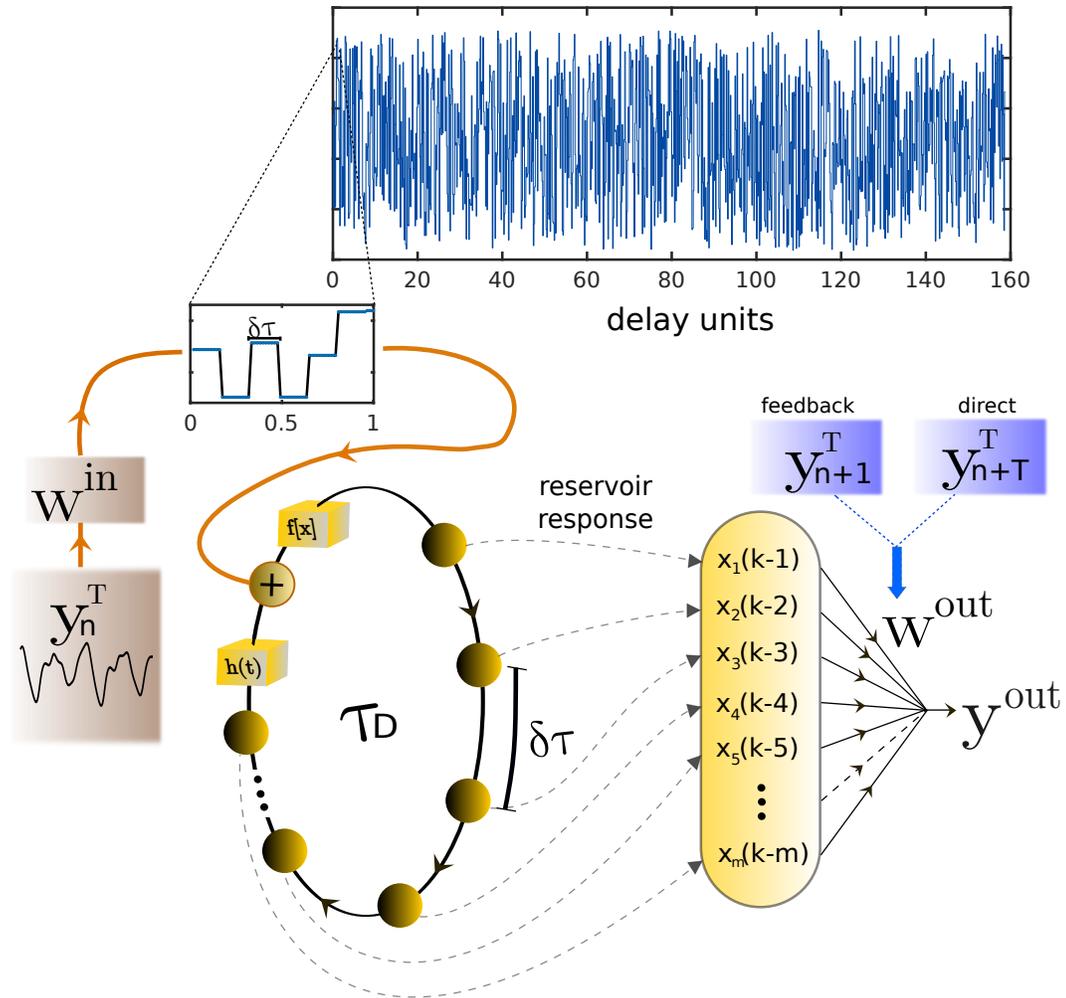


Figure 4.14: Ring-like topology of the nodes performed by the optoelectronic nonlinear circuit, where $f(x)$ constitutes the nonlinear activation function and $h(t)$ denotes the systems impulse response. The input layer is constituted by the input y_n^T and the mask W^{in} . The output layer receives the information as one-step time-delayed time series, $M^x = \{x_1(k-1), x_2(k-2), x_3(k-3), \dots, x_m(k-m)\}$.

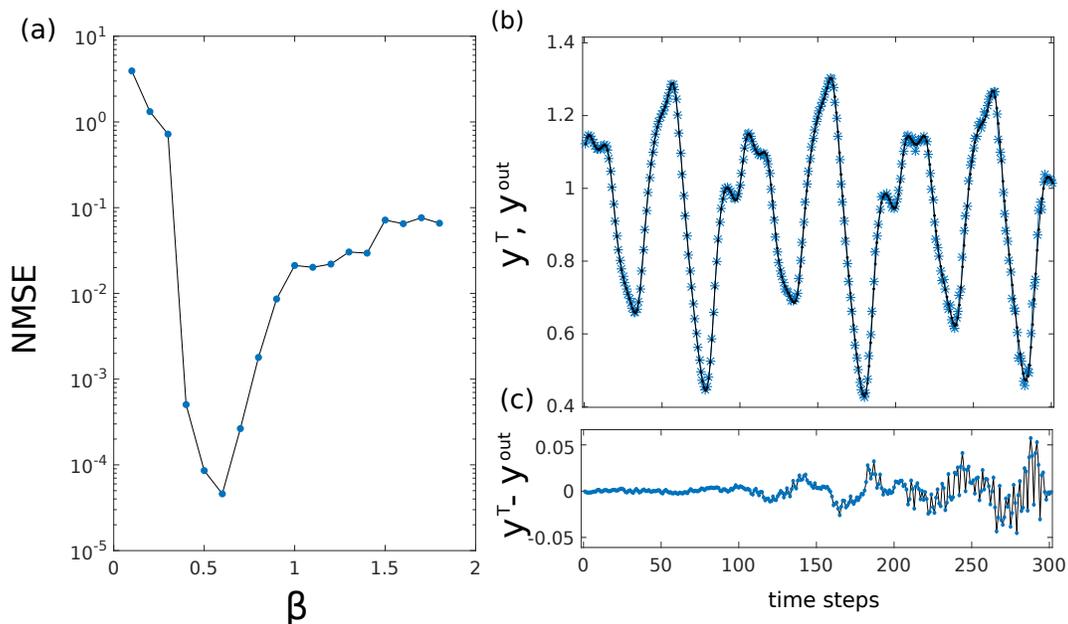


Figure 4.15: Lowpass Ikeda-based feedback predictor. (a) Prediction error (NMSE) of 85 MG steps, (b) 300 steps in the future prediction of MG, with signals y^{out} (blue stars) and target y^T (black dots), and (c) the error between predicted and target signals, $(y^{out} - y^T)$, for $\beta = 0.6$.

input signal $u_{in} = [W^{in} \cdot y_n^T]p_{\delta\tau}$ is composed by the multiplication between the original sequence y_n^T and the input weights vector W^{in} , where each value of y_n^T will multiply W^{in} . After such operations, each resultant vector is time multiplexed via $p_{\delta\tau}$, in order to excite the nodes in the network (see procedure introduced in Sec. 1.4.7).

In practice, we utilize $m = 1000$ nodes in total, which are distributed in sets of $R/\delta\tau$ nodes per delay interval, where $\delta\tau = 10$. The input layer consists in a input signal u_{in} amplified by a factor $\alpha = 0.8$, composed by an input weights vector W^{in} with $m = 1000$ randomly distributed elements in $[-1, 1]$ (generated using the MATLAB routine `rand`); and y_n^T comes from the MG system Eq. (1.7). The multiplexed input sequence u_{in}^i , associated to the i th value of y_n^T (with $i = 1, \dots, n$), consists in approximately 158 delay units of our DRC. Each multiplexed sequence is injected in an independent realization of the Ikeda equation. It means that, if the training step is performed by using a set of $k = 5000$ values of the MG sequence, then we inject independently 5000 values of the MG sequence in 5000 different realizations of the Ikeda equation.

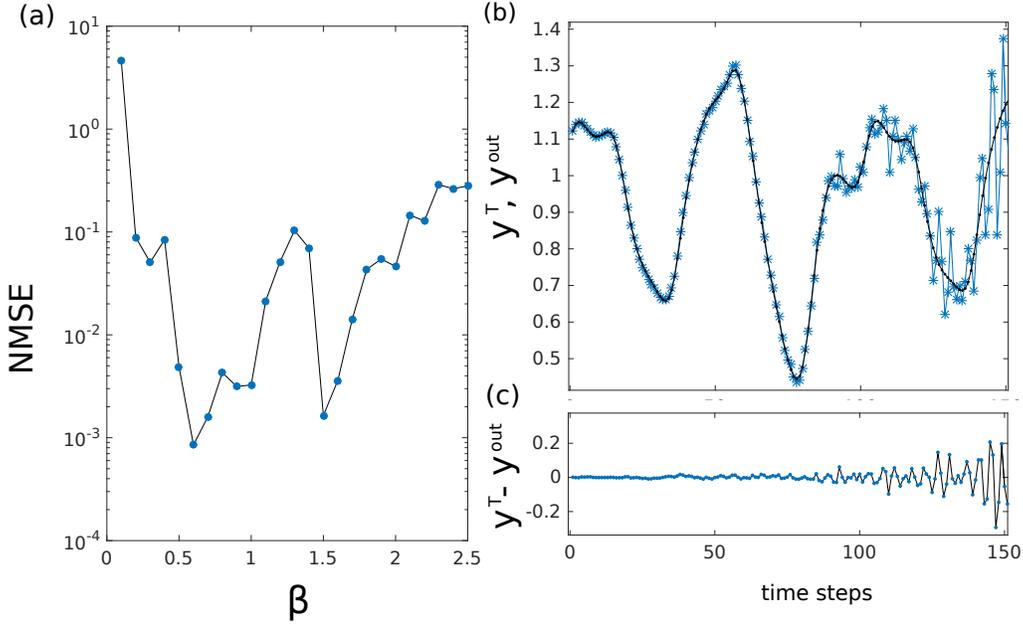


Figure 4.16: Bandpass Ikeda-based feedback predictor. (a) Prediction error (NMSE) of 85 MG steps, (b) 150 steps in the future prediction of MG, with signals y^{out} (blue stars) and target y^T (black dots), and (c) the error between predicted and target signals, $(y^{out} - y^T)$, for $\beta = 0.6$.

That is to say, we inject each value of y_n^T in a multiplexed signal corresponding to 158 delay units of our DRC, and we repeat this procedure 5000 times. The node responses are collected in the matrix $M^x = \{x_1(k-1), x_2(k-2), x_3(k-3), \dots, x_m(k-m)\}$ with dimension $m \times (k-m)$, where each $x_i(j-i)$ is a time series.

Next, we utilize all node responses M^x and a teacher signal y_{n+1}^T to train the network for prediction. The output y^{out} of the entire system is calculated by $y^{out} = W^{out} \cdot M^x$, by optimizing $W_{op}^{out} = \min |W^{out} \cdot M^x - y_{n+1}^T|$, using regression via S.V.D, see Sec. 1.4.8. Until this point, we have investigated just one model for prediction, which is a *feedback* predictor, capable to iterate a single-step prediction several times to get long-term prediction performances. This iterative model requires to connect the predicted output y^{out} with the input y^T , updating the system recurrently with the last predicted value. Now, we will consider a second model defined as *direct* predictor. Such model implies to set a prediction horizon T in the learning step, time shifting the target T time steps in the future, y_{n+T}^T , respect to the input y_n^T . In this case we

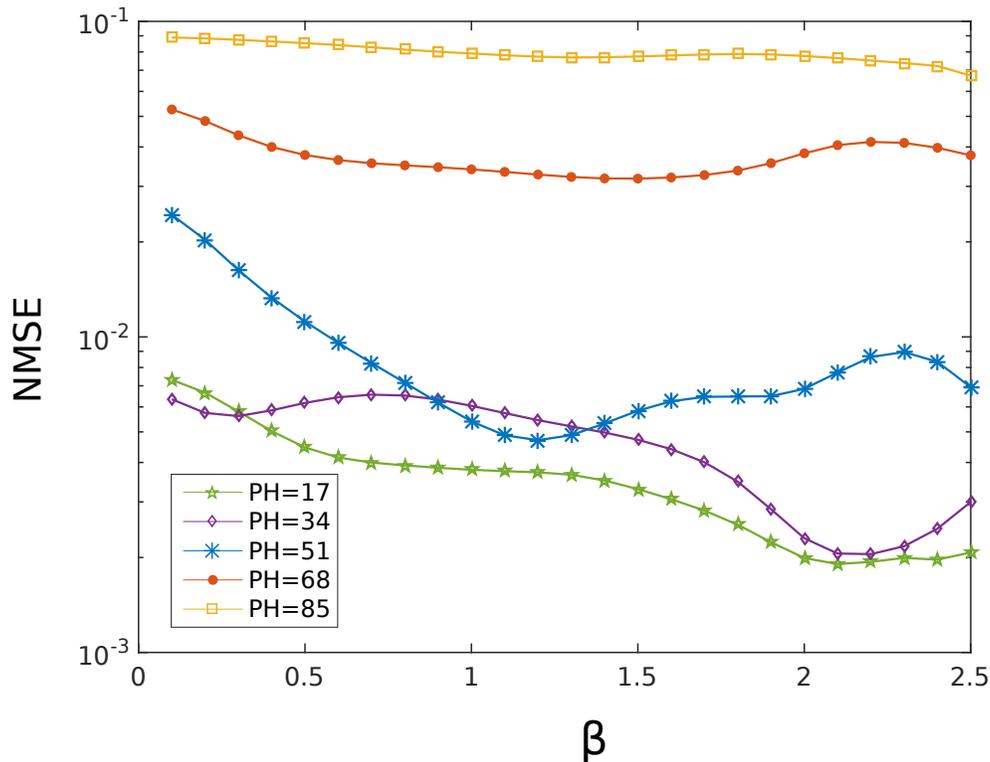


Figure 4.17: Lowpass Ikeda-based direct predictor. Prediction error (NMSE) of MG for prediction horizons $\text{PHs} = \{17, 34, 51, 68, 85\}$.

optimize $W_{op}^{out} = \min |W^{out} \cdot M^x - y_{n+T}^T|$. Then, the testing step does not require the output to be feedback into the system as in the feedback predictor. Consequently, the network is not left running freely for prediction, it just predict T steps in the future all at once. The advantage of using this second model is that it can be easily implemented in hardware such as the electro-optic oscillator introduced in Sec. 1 of this Chapter.

Feedback prediction

The first feedback predictor is build by using the lowpass Ikeda system at $\phi_0 = -0.6\pi$ for feedback long-term prediction of the MG system. To enable such prediction, we disconnect the teacher after the step $n = 5000$ and set y^{out} as input, in order to leave the network running freely 85 steps (five times the time delay of the MG system). In Fig. 4.15(a), the NMSE shows that the best

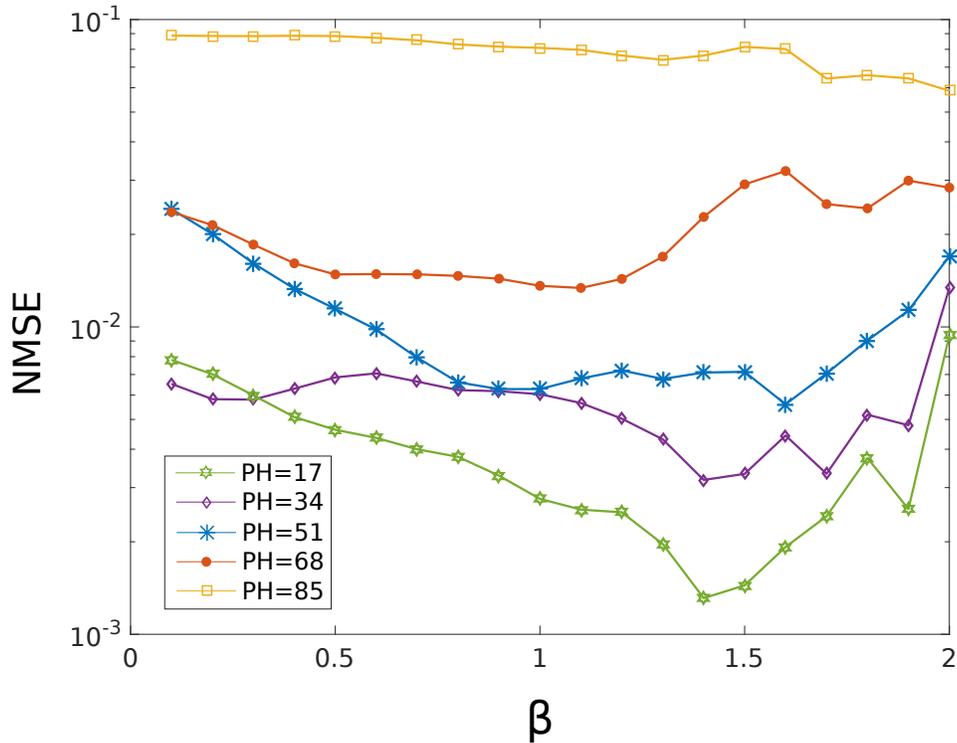


Figure 4.18: Bandpass Ikeda-based direct predictor. Prediction error (NMSE) of MG for prediction horizons PHs= {17, 34, 51, 68, 85}.

long-term prediction performance is found at $\beta = 0.6$ with $\text{NMSE} \simeq 4.6 \times 10^{-5}$. Under this condition, we left the lowpass Ikeda running freely until it predicts 300 steps. The outcome of this test is illustrated in Fig. 4.15(b), where signals y^{out} (blue stars) and target y^T (black dots) are directly compared. Figure 4.15(c) shows how the error between predicted and target signals, $(y^{out} - y^T)$, changes overtime. Here, the smallest errors are found for the first 100 steps of the prediction, but in general we can still predict until 300 steps in the future (which is around eighteen times the time delay of the MG system) with a relative small error (NMSE of 0.04).

The same experiment was repeated for the bandpass Ikeda, with which we build our second predictor, the results are shown by Fig. 4.16. The best prediction performance is found at $\beta = 0.6$ with $\text{NMSE} \simeq 8.5 \times 10^{-4}$, corresponding to 85 steps prediction, see Fig. 4.16(a). A second peak of good performance is found at $\beta = 1.5$ with $\text{NMSE} \simeq 1.6 \times 10^{-3}$. For $\beta = 0.6$, we left the bandpass Ikeda running freely for 150 steps, where signals y^{out} (blue stars) and target

y^T (black dots) are directly compared, see Fig. 4.16(b). The error between predicted and target signals, $(y^{out} - y^T)$, are small for the first 80 steps, but in general we predict until step 150 in the future with a NMSE of 0.06, see Fig. 4.16(c).

Direct prediction

In our second prediction paradigm, we build direct predictors whose performances we can compare with their feedback counterparts. We keep exactly the same topology, input properties and parameters in the direct system as well. Here, we only change the training and testing procedures. In the design of direct predictors, the prediction horizon (PH) has to be defined a-priori in the learning step. In contrast, feedback predictors a-posteriori define their prediction capabilities in the testing step. As a common element, both predictors establish their PHs when the output strongly diverge respect to the target.

We investigate both, lowpass and bandpass Ikeda-based direct predictors with $\phi_0 = -0.6\pi$, by setting various PHs= $\{17, 34, 51, 68, 85\}$, and evaluate the prediction performance for varying β . In Fig. 4.17, the NMSE as function of β for the lowpass Ikeda system is shown, where lower errors are found for the range $\beta \in (1.5, 2)$ for PHs= $\{51, 68, 85\}$, and $\beta \in (1.5, 2.5)$ for PHs= $\{17, 34\}$. We find similar behaviour in the bandpass Ikeda system, see Fig. 4.18. However, the lower errors for the bandpass oscillator are all found for the range $\beta \in (1, 2)$ for all PHs. In general, increasing the PH worsens the performance of our predictors, i.e. the NMSE is almost monotonously increasing with the increment of the PH. Good prediction performances have been found for values of β close to the bifurcation points, which are $\beta \simeq 2.3$ for the lowpass, and $\beta \simeq 1.45$ bandpass Ikeda systems.

A comparison between predictors lead us to conclude that feedback predictors perform better long-term predictions than their direct counterparts. This is related to the fact that direct predictors are not iterative, they cannot generate internal dynamics as feedback predictors do. As introduced in Chapter 2, iterative models require a feedback between the predicted output and the input, updating the system recurrently with the last predicted value. There are causal relationships formed between the predictor's components, coming from the autonomous storage-and-update of information. The stored and new information accumulated in an autonomous feedback loop implies memorization, with which it is possible to enhance long-term prediction performances [43]. We can therefore expect good long-term prediction performances far from the edges of bifurcation, as shown by Fig. 4.15(a). Therefore, our non-feedback

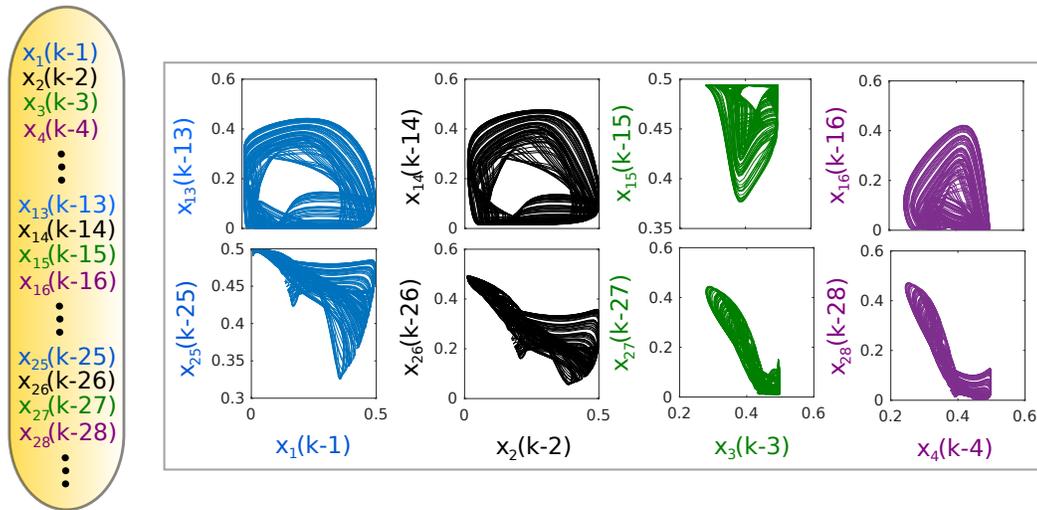


Figure 4.19: First panel to the left, shows the way how to select the node responses to uncover the embedded attractor. Second panel to the right shows the reconstructed dimensions of the MG attractor. The node responses come from the perturbed lowpass Ikeda, but similar objects were found for the bandpass Ikeda.

predictors, unable to generate autonomous internal dynamics, seems to harness the most the high excitability features of the Ikeda systems' edges of bifurcation to enable long-term prediction.

Finally, we can look inside our networks to understand if there are evidences of Takens-like embedding from node responses, as introduced in Chapter 2. Additionally, we search if different versions of the MG attractor found in the Ikeda system could provide part of the required structure to build a predictor, as it was described in Chapter 2. The first set of nodes lagged at the values of the Takens scheme is $\{x_1(k-1), x_{13}(k-13), x_{25}(k-25), x_{37}(k-37)\}$. The second set of nodes is $\{x_2(k-2), x_{14}(k-14), x_{26}(k-26), x_{38}(k-38)\}$. From the third to the last set, the procedure to obtain sets of nodes lagged at the values of the Takens scheme follows the same structure. This procedure let us identify if the delay system is creating modified versions of the originally injected object as we could find in the rRNNs.

We show some 2D projections of the embedded MG attractor in Fig. 4.19, where some of them are qualitatively comparable to the reconstructed MG attractors from the Takens scheme, see Fig. 1.9, and others are topologically equivalent but look more distorted and folded. In the case of rRNNs, the ran-

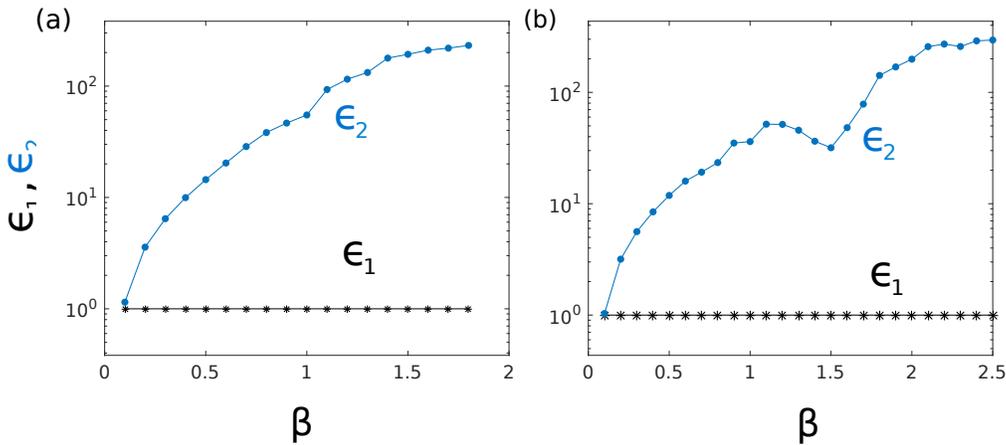


Figure 4.20: Maximum and minimum boundaries $\{\epsilon_1, \epsilon_2\}$ for (a) lowpass and (b) bandpass Ikeda-based predictors.

dom mapping was found to play an important role to create nearest neighbors for prediction, however the Ikeda system creates these neighbors in a different way. The neighbors come from distortions on the MG attractor related to the nonlinear damping of the Ikeda oscillator. Since the nonlinearity appears in the dissipation coefficient of the equation, the oscillator tries to maintain the constant amplitude of the stationary state, even if there is an external perturbation. This inertia-like behavior of the Ikeda oscillator distorts the incoming information, and binds it to its relative basin of attraction. Such boundary structures perform vicinities from where neighbors merge.

The ensemble of various embedded attractors in the lowpass Ikeda can be quantitatively investigated through the estimation of the limits $\{\epsilon_1, \epsilon_2\}$ of interstate distances in the network. Following Sec. 2.2.1, we determine $\{\epsilon_1, \epsilon_2\}$ for several values of β . Figure 4.20(a) shows that $\epsilon_1 \approx 1$ remains constant for all β . Therefore, embedded versions of the MG attractor are found to have very small minimal interstate distances for all β . The evolution of ϵ_2 shows that the maximal interstate distance is monotonically increasing with β . In fact, for $\beta = 0.2$ the limit ϵ_2 is already close to one, then long-term prediction is in fact theoretically possible according to the results presented in Chapter 2. However, good long-term predictions start at $\beta = 0.4$ and $\beta = 0.5$ for the lowpass and bandpass-based feedback predictors, respectively. In these cases, the limit $\epsilon_2 \simeq 10$ corresponds to the case where the versions of the MG attractor allow a proper scan of the MG's neighborhood. As we conclude in Chapter 2, currently we are not able to deduce how the neighbors are distributed in the

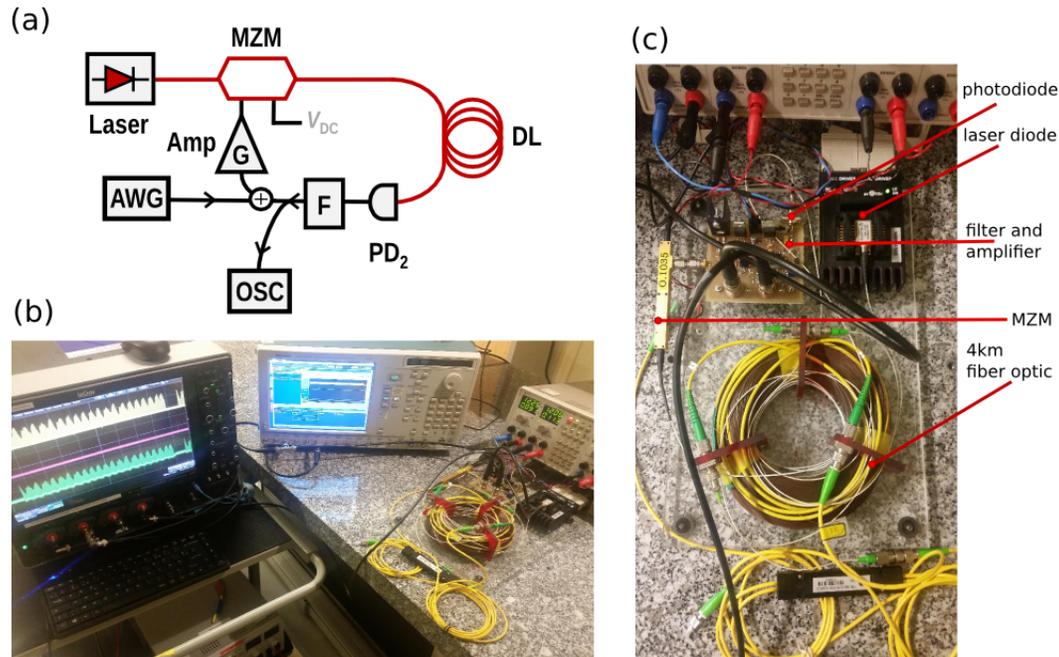


Figure 4.21: (a) Schematic diagram of the photonic direct predictor. MZM: Mach-Zehnder Modulator; PD: Photodiode; DL: Delay-Line; F: lowpass or Bandpass Filter; Amp: RF amplifier; AWG: arbitrary waveform generator; OSC: oscilloscope. (b) Picture of the experimental setup, showing the devices from left to right, consists in the oscilloscope, AWG, and the delay electro-optic circuit. (c) Zoom on the circuit, showing individual elements of Ikeda photonic implementation.

network to enable prediction, they could build clusters of neighbors which are not randomly scattered along the state space region where the attractors are. Therefore, the information required to complete the chaotic attractor model under study is limited. Good prediction results (see Fig. 4.15(a)) seems to stop when ϵ_2 is sufficiently large ($\epsilon_2 \sim 1 \times 10^2$), then interstate distances of embedded attractors are considered far out of the nearest neighbourhood.

The same experiment was repeated for the bandpass Ikeda, and the results are shown by Fig. 4.20(b), where the estimated limits $\{\epsilon_1, \epsilon_2\}$ have a comparable evolution as function of β , until $\beta = 1.3$. In this point the values of ϵ_2 start decreasing with the increment of β , disrupting therefore the monotonous increment performed by ϵ_2 . The limit $\epsilon_2 < 1 \times 10^2$ is coincidental with good

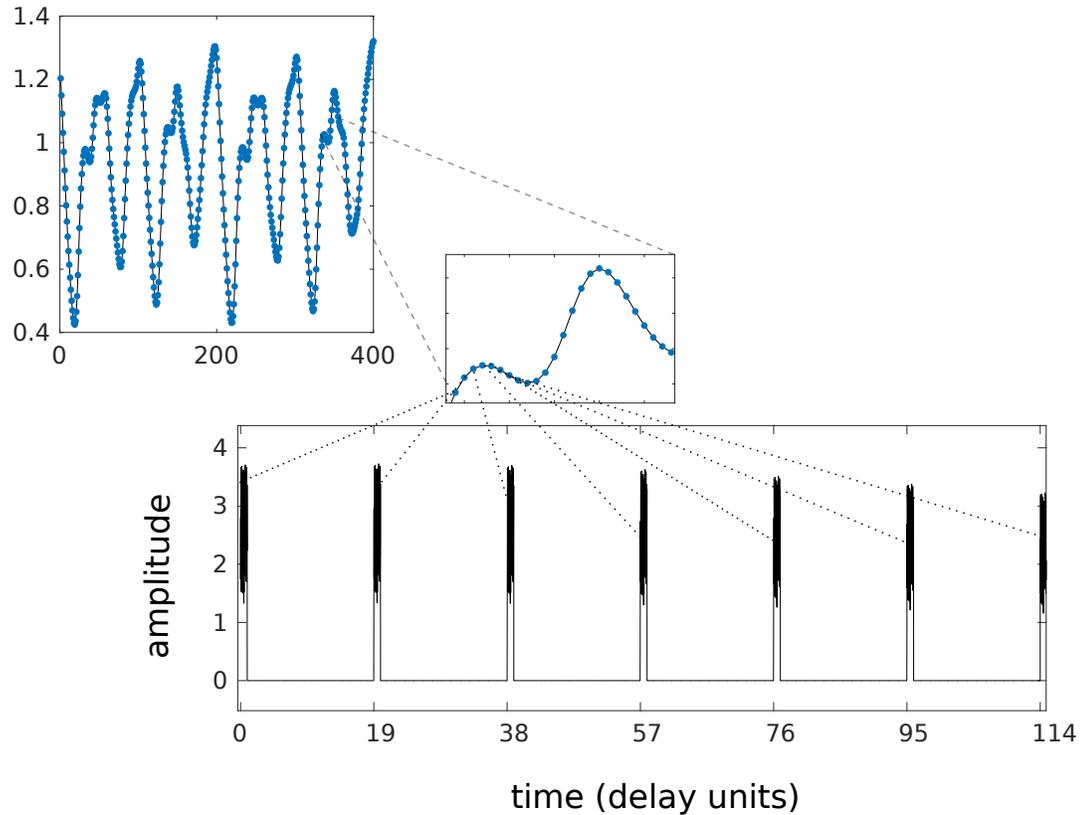


Figure 4.22: Illustration showing a set of 400 points from the MG system, then a zoom in a subset of it, from where we extract datapoints to be masked and distributed along all the input trace.

long-term prediction performances in Chapter 2, shown by Fig. 4.16(a).

4.2.3 Photonic reservoir computing for direct prediction

The experimental implementation of the photonic reservoir computing (PRC) concept is based on the electro-optic oscillator introduced in section 4.1 [7, 8, 141, 142]. Figure 4.21(a) shows the schematic diagram of the full circuit for prediction. Here, we add a connection to an arbitrary waveform generator (AWG), through which the information is injected into the electro-optical oscillator. Then, after a round-trip the processed information will be recorded by the oscilloscope. Also, the filter (F) can be set either lowpass or bandpass,

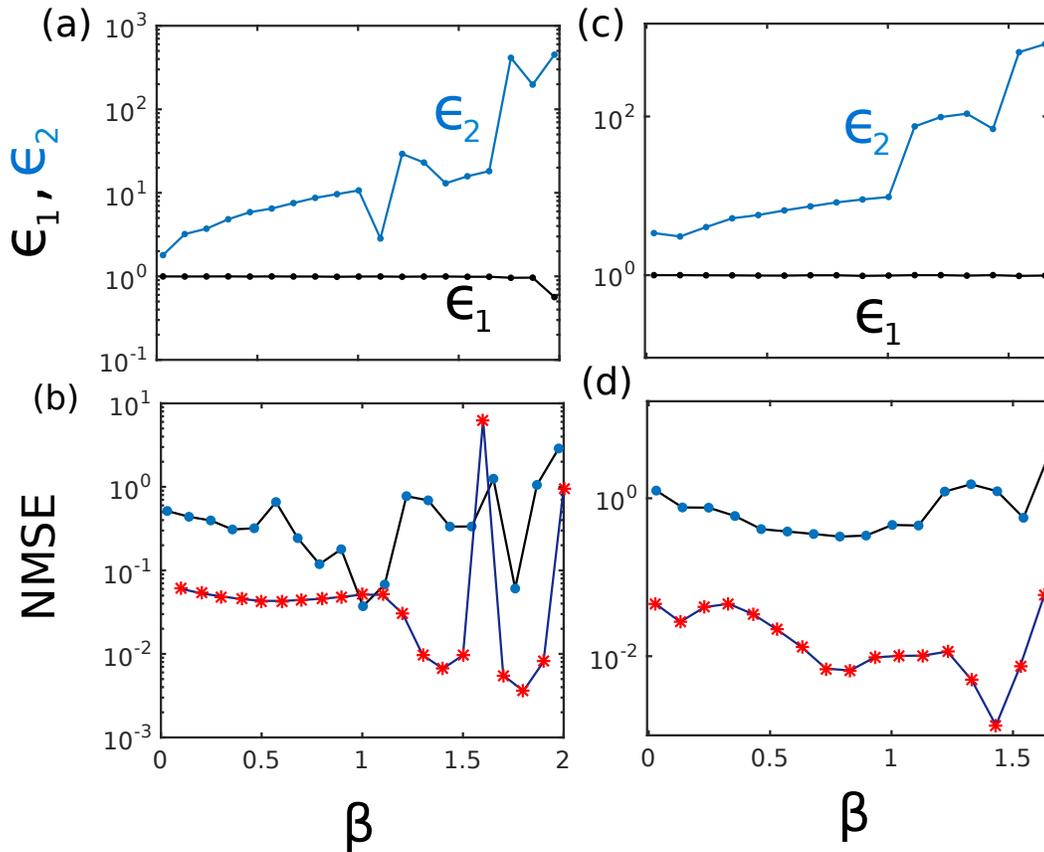


Figure 4.23: Both panels contain experimental maximum and minimum boundaries $\{\epsilon_1, \epsilon_2\}$ on the top, and experimental (blue dots) and numerical (red stars) prediction error (NMSE) of 6 MG steps on the bottom, for (a,b) lowpass and (c,d) bandpass Ikeda-based direct predictor.

keeping the same high cutoff frequency. An illustrative picture of our setup is shown by Fig. 4.21(b), where the devices from left to right consist in the oscilloscope, AWG, and the delay electro-optic circuit. A zoom on the circuit can be seen in Fig. 4.21(c), where we can find the individual elements which model and solve the Ikeda DDE: photodiode, laser diode, MZM, filter, amplifier and 4 km fiber optic.

Up to this point, we have investigated the operating point which leads to the best prediction performances in the Ikeda-based feedback and direct predictors. However, in our experiments we define the operating point at $\phi_0 = -\pi/2$, allowing us to work at the extreme of the nonlinear function. Such operating

point is the most suitable to be used under noisy conditions, which are expected in experiments. In fact, the MZM is a device that experiences several drifts that can provide additional instabilities leading to bad processing performances. Consequently, it is important to work with an operating point which can easily be identified once it is spontaneously changed by the drifts of the MZM.

In the previous approach, we injected each single value of u_{in} in approximately 158 delay units of our DRC. It means that if u_{in} consists in 5000 values, we have to repeat 5000 times the injection of those 158 delay units carrying each single value of the sequence u_{in} . In our current experimental setup, such task is far time consuming and risky, because we have to repeat injection of a sequence of 5000 values from the MG system in 5000 independent realisations of the same experiment. Then, drifts of the MZM can unset the whole experiment in ways that we cannot handle with our current resources. With the aim to avoid these issues, the training dataset has to be all injected at once in one single realisation of the experimental Ikeda systems. Yet, this solution leads us to the problem that we would have to inject a large amount of data in a single realization of the same experiment. Therefore, our first step is to reduce the amount of information to be injected.

For that, a new connectivity of the PRC is defined by setting $\delta\tau = 5$. With this value defining the space between network nodes, we increase the amount of coordinates to be used per delay in our reservoir. Also, we use a smaller amount of nodes, $m = 100$, and input connections via weights vector W^{in} with 100 randomly distributed elements in $[-1, 1]$. Then, the multiplexed input sequence u_{in}^i associated to the i th value of y_n^T , consists in approximately 8 delay units of our DRC. The training step is performed by using a learning set of $k = 1000$ MG datapoints.

In the next step, we have to emulate k independent realisations of the Ikeda systems. Then, between two different values $\{u_{in}^i, u_{in}^{i+1}\}$ we leave large blank spaces. By doing this, it is possible to untangle and separate two consecutive input values as if they were being processed in independent realisations of the Ikeda equations. Figure 4.22 shows an illustrative example of this procedure. Here, a set of 400 points from the MG system is chosen, then we zoom into a subset of this sequence, from where we extract datapoints to be masked and distributed along all the input trace. By using this procedure, we are capable to process and record 1000 values from the MG time series, corresponding to the full length of the training sequence in our experiment. The information is injected via the AWG and recovered by the oscilloscope both with a sampling frequency of 25 Ms/s.

The experiment is carried out by the injection of two sets in two different

experimental steps. The first one is the set for the training, and the second is for the testing. The testing set is extracted from the same MG system but 100 steps shifted, it means that if the temporal domain of the training signal is in $[1, 1000]$, then the testing signal's domain is in $[101, 1100]$. The prediction horizon defined for this set of nodes and training values is $T = 6$. The aim therefore is to predict points from 1100 to 1105 of the MG signal. The training step is done offline.

We start the analysis by estimating experimental limits $\{\epsilon_1, \epsilon_2\}$ for the low-pass Ikeda oscillator. Figure 4.23(a) shows the evolution of the limits as β increases. It can be seen that ϵ_1 remains close to the unity, indicating the presence of small minimal interstate distances from the embedded MG versions in the PRC. On the other hand, ϵ_2 exhibits a more complex behavior. In Fig. 4.23(b), we can find two separated sets with small NMSE around $\beta = \{1, 1.7\}$ in the experiment (blue dots) as well as in the numerical simulation (red stars). The best experimental prediction performance was found in two regions, one around $\beta = 1$ where NMSE= 0.03, and the second around $\beta = 1.7$ where NMSE= 0.06.

The same analysis was implemented on the bandpass Ikeda-based predictor. $\{\epsilon_1, \epsilon_2\}$ were estimated as functions of β , see Fig. 4.23(c). The here presented experimental limits give $\epsilon_1 \approx 1$ for all β , and ϵ_2 increasing with some irregularities. Contrary to the lowpass case, this predictor contains just one region associated to good performance, that is around $\beta = 1.4$ in the experiment (blue dots) as well as in the numerical simulation (red stars). The lower NMSE is 0.3, which is by one order of magnitude worst than for the lowpass-based predictor. Due to the integral component of the bandpass Ikeda system, the oscillator is integrating along all the time series, causing additional distortions to the processed data.

4.2.4 Conclusion

We have investigated the autonomous dynamics of a bandpass delayed nonlinear EO system, and their particular slow-fast solutions, when the nonlinear feedback involves a positive slope between two extrema. Depending on the strength of the nonlinear delayed feedback weighted by the bifurcation parameter β , various solutions are bifurcating. The particular parameter conditions of concern have been recently investigated in the view of Liénard limit cycle. We have further shown that beyond the Liénard limit cycle solution, the Liénard-Ikeda solution can be interpreted as the unidirectional coupling of a conventional (i.e. low pass instead of bandpass) Ikeda dynamics and a Liénard

limit cycle modulating the phase parameter of the Ikeda dynamics. Our result is demonstrated experimentally by a lowpass Ikeda dynamics externally driven by a triangular waveform typically resulting as slow dynamical term from a Liénard system. Experimental and numerical results are found to be in excellent agreement, and results are supported by analytics developed on the original bandpass Ikeda model. Future work will focus on the exploration of such system dynamics for strong β , when the three timescales θ , τ and τ_D are suspected to mutually trigger the chaotic oscillations, thereby inducing a higher complexity in the timescale interactions.

In the second section, we have studied in theory, numerics and experiments the properties of two different predictors (feedback and direct) in a delay oscillator. The predictors were used to perform short and long-term prediction of MG chaotic time series. We have shown that feedback predictors are in fact the best long-term predictors in our framework, by estimating more than 100 steps in the future with low errors. In contrast, direct predictors were found to be more inaccurate for this task, showing relatively high errors for 85 steps prediction. However, direct predictors can be used as short-term predictors since they can be trained to estimate less than 34 steps in the future with small NMSE. As our experimental setup was not suitable to implement feedback prediction paradigm, we could use it for short-term direct prediction. For prediction horizon $T = 6$, we built Ikeda-based direct predictors by using two different filters: lowpass and bandpass with the same high cutoff frequency. The performance of the bandpass predictor was shown to be the less optimum, since the integral term in the bandpass Ikeda oscillator introduced additional noise-like distortions to the processed time trace in our approach.

Chapter 5

Conclusion and perspectives

Conclusions

This work starts with the study of spatial rRNNs based on a toolbox from dynamical systems theory. We have introduced a novel methodology providing a long-missing understanding of how prediction is achieved by rRNNs. Previous approaches typically reduced the required characteristics simply to the memory capacity or comparable measures. Yet, a chaotic signal is determined by the structure of its phase space, which is far more complex than a single parameter. We provide this essential connection and have identified how the complex and high-dimensional interactions within a rRNN allow such systems to predict with excellent performance. We have shown that rRNNs are capable to build a spatial representation of a 1-dimensional chaotic time series. Our finding partially removes the black-box property of rRNNs for prediction.

The consequences of our novel framework are far-reaching. A systematic understanding of the intricate linkage between the rRNN space and the chaotic attractor enables the field of neural networks to pursue new avenues. It allows for the development and crucially an effective evaluation of new training methods. Furthermore, our scheme based on random nonlinear mapping and nearest neighbors identifies the key ingredients for prediction based on high-dimensional nonlinear systems. This approach also allows us to improve the design of our neural network in order to reduce the number of nodes and connections required to solve prediction. The practical implications of our scheme are directly related with the identification of strategies to modify classical computers for prediction by a priori defining external memory access rules.

The next approach was about the investigation of dynamical complexity in multistable rRNNs beyond their fixed point. We therefore demonstrated processes inside a random neural network which allow good prediction performance at parameters where learning in biological neural networks benefits. Based on a rRNN with a periodic nonlinear function, we compare various regions of regular dynamics and highlight their importance of spatial synchronization upon prediction performance, mutual information and the stability of the neural network. Synchronization between nodes plays an essential role,

but it is not sufficient to understand how information processing is successful in a rRNN beyond its fixed point.

We describe a rRNN predicting the future time-steps of the a chaotic trajectory. Our results illustrate the importance of information flow, divergence and the suppression of signal components not present in the training data set. The rRNN's damped autonomous deviation $\sigma_{\alpha=0}$, mutual information MI and maximal Lyapunov exponent can be seen as complexity indicators for interpreting neural networks based on dynamical systems. Other than for the oscillatory state, chaotic responses were not capable to maintain important features of the input dynamic, resulting in low prediction performances.

Finally, we have investigated at first the autonomous dynamics of a delayed nonlinear electro-optic system capable to exhibit slow-fast hybrid trajectories, which are associated to different parts of the system's nonlinear transfer characteristic function. For instance, the neighbourhoods of the function extrema are related to oscillations typical of Ikeda equation. However, if $\phi_0 = -\pi/4$ we can expect to have two Ikeda-like fast-scale solutions coexisting in different regions of the phase space and connected to each other by a large amplitude slow-scale periodic envelop.

In order to generate those solutions, the dynamical system autonomously harness a cubic polynomial fraction of the quartic polynomial fulfilling the role of the MZM's transfer characteristic. This is consequential of the associated energy-based approach, through the equivalence with the general Liénard system. Subsequently, the 2D projection of the reconstructed orbits in the phase space have been found in a finite region of it, due to the limits that the potential function impose to them. From the attractor reconstruction, we discovered spiral-shaped Ikeda-like solutions on the extrema of a huge limit cycle, related to the fact that the high-pass filter in the electro-optic circuit can be understood as a coupling element between two Ikeda solutions.

In the second section of this last part, we have studied the properties of feedback and direct predictors in a delay oscillator, for short and long-term prediction of MG chaotic time series. We have shown that feedback predictors are better for long-term prediction than direct ones. However, direct predictors can be used for short-term prediction, and then easily implemented in our experimental setup. For prediction horizon $T = 6$, we built Ikeda-based direct predictors by using two different filters: lowpass and bandpass with the same high cutoff frequency. The performance of the bandpass predictor was shown to be the less optimum, since the integral term in the bandpass Ikeda oscillator added additional noisy-like distortions to the processed time trace in our approach.

Perspectives

The perspectives for future works can be split into the following two main perspectives:

- Extend the theory to different tasks such as classification. Classification task of spoken digits contains hints about embedding as a part of the procedure to classify any elements. Here, we use cochleagram to enable perceptual separation of sound. The number of frequency channels used depends on the sampling frequency of the pronounced numbers. For a sampling frequency of 12.5 kHz, we usually use 86 filters. An important characteristic of the cochlea is that the energy in the acoustic wave is separated by frequency, and each point in the cochlea will respond best to one frequency. Then, the cochlea maps the frequency content of the signal into a spatial domain, with a very good accordance to the underlying dynamic of the original data. It means that these time-frequency representations of the data are structure preserving, and the mapping from spoken digits to time-frequency is injective. Therefore, the object is embedded before it is processed by the reservoir. That is to say, we already provide a spatial representation of the problem to the reservoir in this task. The problem which merge here is that we are still not capable to find a nearest neighborhood that we can relate to good classification performance, as in the case of prediction.

- In experiments, we aim to build a full analog electro-optic feedback predictor. Until now it is shown that we can use the delay photonic reservoir to solve prediction and classification. In this experiments typically the reservoir layer is experimental, and the other two (input and output) are performed by a computer in a digital environment. Then, the full analog experimental setup has not been performed yet. The next steps will be related to build the input and output layers experimentally. For the delay electro-optic setup, we propose to design optical memories with fiber optics and fiber amplifiers in order to implement the mask and output weights. By doing this, we aim to physically feedback the input and output layers to have at the end a feedback predictor with which perform long-term prediction of chaotic time series.

Bibliography

- [1] S. Ackerman. *Discovering the brain*. The national academies press, 1992.
- [2] R. Rojas. *Neural networks: A Systematic Introduction*. Springer, 1996.
- [3] K. Gurney. *An Introduction to Neural Networks*. CRC Press, 1997.
- [4] W. Maass. *Curr. Opin. Behav. Sci.*, 11:81, 2016.
- [5] H. Jaeger and H. Haas. *Science*, 304:78, 2004.
- [6] W. Maass, T. Natschlaeger, and H. Markram. *Neural Comput.*, 14:2531, 2002.
- [7] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer. *Nat. Commun.*, 2:468, 2011.
- [8] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer. *Opt. Express*, 20:3241, 2012.
- [9] E. Ott. *Chaos in Dynamical Systems*. Cambridge University Press, 1993.
- [10] S. H. Strogatz. *Nonlinear Dynamics and Chaos*. Westview Press, 2000.
- [11] H. G Schuster. *Deterministic chaos: An Introduction*. Wiley-VCH Verlag GmbH, 2005.
- [12] M. Gossel. *Nonlinear time-discrete systems*. Springer-Verlag, 1982.
- [13] A. Pikovsky, M. Rosenblum, and J. Kurths. *Synchronization*. Cambridge University Press, 2001.
- [14] E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20:130–141, 1963.
- [15] O. E. Rossler. An equations for continuous chaos. *Phys. Lett.*, 57A:397–398, 1976.
- [16] O. E. Rossler. Chaotic behaviour in simple reaction systems. *Z. Naturforsch*, 31A:259–274, 1976.

-
- [17] G. V. Demidenko, V. A. Likhoshvai, and A. V. Mudrov. *Differential Equations*, 45:33, 2009.
- [18] M. C. Mackey and L. Glass. Oscillation and chaos in physiological control systems. *Science*, 197:287–289, 1977.
- [19] K. Ikeda. *Opt. Commun.*, 30:257, 1979.
- [20] K. Ikeda, H. Daido, and O. Akimoto. *Phys. Rev. Lett.*, 45:709, 1980.
- [21] F. Takens. *Detecting strange attractors in turbulence*. Dynamical Systems and Turbulence, Lecture Notes in Mathematics, 1981.
- [22] P. F. Dunn. *Measurement and data analysis for engineering and science*. McGraw-Hill, 2005.
- [23] J. P. Eckmann and D. Ruelle. *Rev. Mod. Phys.*, 57:617, 1985.
- [24] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. *Phys. Rev. Lett.*, 45:712, 1980.
- [25] H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis*. Cambridge University Press, 1997.
- [26] S. Sato, M. Sano, and Y. Sawada. *Prog. Theor. Phys.*, 77:1, 1987.
- [27] M. T. Rosenstein, J. J. Collins, and C. J. De Luca. *Physica D*, 65:117, 1993.
- [28] C. E. Shannon and W. Weaver. *The mathematical theory of communication*. University of Illinois Press, 1949.
- [29] W. McCulloch and W. Pitts. *Bull. Math. Biophys.*, 5:115, 1943.
- [30] I. Khurana. *Medical Physiology for Undergraduate Students*. Elsevier India, 2011.
- [31] J. Snow. *Brainlike computers are a black box. Scientists are finally peering inside*. *Science*, 2017.
- [32] C. A. Hidalgo M. C. Gonzalez and A. L. Barabasi. *Nature*, 453:779, 2008.
- [33] C. A. Hidalgo A. L. Barabasi P. Wang, M. C. Gonzalez. *Science*, 324:1071, 2009.

- [34] V. Mnih et al. *Nature*, 518:529533, 2015.
- [35] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. De Jesús. *Neural Network Design*. Martin Hagan, 2014.
- [36] Y. Tenne and C. K. Goh. *Computational Intelligence in Expensive Optimization Problems*. Springer-Verlag Berlin and Heidelberg GmbH and Co. K, 2012.
- [37] Y. LeCun, L. Bottou, Y. Bengio, P., and Haffner. *Proc. IEEE*, 86:22782324, 1998.
- [38] Y. Bengio, P. Simard, and P. Frasconi. *Neural Netw.*, 5:157166, 1994.
- [39] Y. LeCun, Y. Bengio, and G. Hinton. *Nature*, 521:436, 2015.
- [40] C. G. Langton. *Physica D*, 42:12, 1990.
- [41] T. Natschlager, N. Bertschinger, and R. Legenstein. *At the edge of chaos: Realtime computations and self-organized criticality in recurrent neural networks*. Advances in Neural Information Processing Systems, 2005.
- [42] J. J. Hopfield. *Proceedings of the National Academy of Sciences of the USA*, 79:2554, 1982.
- [43] A. S. Weigend and N. A. Gershenfeld. *Time series prediction: Forecasting the future and understanding the past*. Westview Press, 1993.
- [44] K. L. Priddy and P. E. Keller. *Artificial neural networks: an introduction*. SPIE, 2005.
- [45] S. Amari. *Biol. Cybern.*, 27:77, 1977.
- [46] G. M. Scott and W. H. Ray. *J. Process Control*, 3:163, 1993.
- [47] D. Svozil, V. Kvasnicka, and J. Pospichal. *Chemom. Intell. Lab. Syst.*, 39:43, 1997.
- [48] J. Schmidhuber. *Neural Netw.*, 61:85, 2015.
- [49] H. Sompolinsky, A. Crisanti, and H. J. Sommers. *Phys. Rev. Lett.*, 61:259, 1988.
- [50] M. Stern, H. Sompolinsky, and L. F. Abbott. *Phys. Rev. E*, 90:062710, 2014.

- [51] A. M. Bruckstein, D. L. Donoho, and M. Elad. *SIAM Review*, 51:34, 2009.
- [52] S. Ganguli and H. Sompolinsky. *Ann. Rev. Neurosci.*, 35:485, 2012.
- [53] B. Babadi and H. Sompolinsky. *Neuron*, 83:1213, 2004.
- [54] M. Schottdorf, W. Keil, D. Coppola, L. E. White, and F. Wolf. *PLoS Comput. Biol.*, 11:e1004602, 2015.
- [55] R. MacGregor. *Neural and Brain Modeling*. ACADEMIC PRESS, 1987.
- [56] O. Sporns. *Front. Comput. Neurosci.*, 5:5, 2011.
- [57] A. Gutiérrez and S. Marco. *Biologically Inspired Signal Processing for Chemical Sensing*. Springer, 2009.
- [58] S. J. C. Caron, V. Ruta, L. F. Abbott, and R. Axel. *Nature*, 497:113, 2013.
- [59] M. C. Soriano, S. Ortin, D. Brunner, L. Larger, C. R. Mirasso, I. Fischer L, and L. Pesquera. *Opt. Express*, 21:12, 2013.
- [60] F. T. Arecchi, G. Giacomelli, A. Lapucci, and R. Meucci. *Phys. Rev. A*, 45:4225, 1992.
- [61] Y. Maistrenko L. Larger, B. Penkovsky. *Phys. Rev. Lett.*, 111:054103, 2013.
- [62] Y. Maistrenko L. Larger, B. Penkovsky. *Nat. Comm.*, 6:7752, 2015.
- [63] L. Larger, A. Baylon-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot. *Opt. Express*, 7:011015, 2017.
- [64] S. H. Strogatz. *Physica D*, 143:1, 2000.
- [65] D. S. Watkins. *Fundamentals of Matrix Computations*. John Wiley and Sons, 2004.
- [66] M. Zarandi, M. Zarinbal, N. Ghanbari, and I. Turksen. *Information Sciences*, 222:213, 2013.
- [67] A. Celikyilmaz and I. B. Turksen. *Information Sciences*, 177:5163, 2007.
- [68] I. B. Turksen. *Applied Soft Computing*, 8:1178, 2008.

- [69] S. Goudarzi, S. Jafari, M. H. Moradi, and J. C. Sprott. *Phys. Lett. A*, 380:696, 2016.
- [70] N. Wiener. *Nonlinear Problems in Random Theory*. Wiley and MIT Press, 1958.
- [71] V. Volterra. *Theory of Functionals and of Integrals and Integro-Differential Equations*. Dover Publications, 1959.
- [72] J. D. Farmer and J. J. Sidorowich. *Phys. Rev. Lett.*, 59:845, 1987.
- [73] A. K. Alparslan, M. Sayar, and A. R. Atilgan. *Phys. Rev. E*, 58:2640, 1998.
- [74] D. Kugiumtzis, O. C. Lingjærde, and N. Christophersen. *Phys. D*, 112:344, 1998.
- [75] H. Whitney. *Differentiable manifolds*. Ann. of Math., 1936.
- [76] M. Lukosevicius and H. Jaeger. *Computer Science Review*, 3:127, 2009.
- [77] R. Cox, F. Dabek, F. Kaashoek, J. Li, and R. Morris. *SIGCOMM Comput. Commun. Rev.*, 34:113, 2004.
- [78] P. Indyk and R. Motwani. *Approximate nearest neighbors: towards removing the curse of dimensionality*. ACM Press, 1998.
- [79] P. Frankl and H. Maehara. *J. Combin. Theory B*, 3:355, 1988.
- [80] W. B. Johnson and J. Lindenstrauss. *Conference in modern analysis and probability*. Contemporary Mathematics, Amer. Math. Soc., 1984.
- [81] S. Dasgupta and A. Gupta. *Random Struct. Alg.*, 22:60, 2002.
- [82] D. Sivakumar. *Algorithmic derandomization using complexity theory*. Proc. 34th Annu. ACM symp. Theory of computing, Canada, 2002.
- [83] J. B. Tenenbaum, V. de Silva, and J. C. Langford. *Science*, 290:2319, 2000.
- [84] R. Hecht-Nielsen. *Computational Intelligence: Imitating Life*. IEEE Press, 1994.

- [85] E. Bingham and H. Mannila. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001.
- [86] J. R. Buck, M. M. Daniel, and A. C. Singer. *Computer Explorations in Signals and Systems Using MATLAB*. Prentice Hall, 2002.
- [87] M. Small. *Applied Nonlinear Time Series Analysis: Applications In Physics, Physiology And Finance*. Wspc, 2005.
- [88] T. Schreiber. *Efficient Neighbor Searching in Nonlinear Time Series Analysis*. 1996.
- [89] A. Longtin. *J. Stat. Phys.*, 70:309, 1993.
- [90] D. J. Christini and J. J. Collins. *Phys. Rev. Lett.*, 75:2782, 1995.
- [91] R. M. Enoka. *Neuromechanics of human movement*. Human Kinetics, 2015.
- [92] B. Tirozzi, D. Bianchi, and E. Ferraro. *Introduction to computational neurobiology and clustering*. World Scientific Publishing Company, 2007.
- [93] A. Garfinkel, M. L. Spano, W. L. Ditto, and J.N. Weiss. *Science*, 257:1230, 1992.
- [94] S J. Schiff, K. Jerger, D H. Duong, T. Chang, M L. Spano, and W. L. Ditto. *Nature*, 370:615, 1994.
- [95] K. Lehnertz, C. E. Elger, J. Arnhold, and P. Grassberger. *Chaos in Brain?* World Scientific Publishing Company, 2000.
- [96] E. M. Harth, B. Beek, T. J. Csermely, and R. D. Lindsay. *Theor. Biol.*, 26:93, 1970.
- [97] G. Wainrib and J. Touboul. *Phys. Rev. Lett.*, 110:118101, 2013.
- [98] J. Fell and N. Axmacher. *Nat. Rev. Neurosci.*, 12:105, 2011.
- [99] P. Konig, A. K. Engel, and W. Singer. *Trends Neurosci.*, 19:130, 1996.
- [100] R. Azouz and C. M. Gray. *Proc. Natl Acad. Sci. USA*, 97:8110, 2000.
- [101] G. Daoudal and D. Debanne. *Learn. Mem.*, 10:456, 2003.

- [102] W. H. Miltner, C. Braun, M. Arnold, H. Witte, and E. Taub. *Nature*, 397:434, 1999.
- [103] S. Weiss and P. Rappelsberger. *Brain Res. Cogn. Brain Res.*, 9:299, 2000.
- [104] J. Fell, E. Ludowig, T. Rosburg, N. Axmacher, and C. E. Elger. *Neuroimage*, 43:410, 2008.
- [105] K. Benchenane, A. Peyrache, M. Khamassi, P. L. Tierney, Y. Gioanni, F. P. Battaglia, and S. I. Wiener. *Neuron*, 66:921, 2010.
- [106] R. Singh, S. N. Menon, and S. Sinha. *Sci. Rep.*, 6:22074, 2016.
- [107] D. Q. Wei and X. S. Luo. *EPL*, 78:68004, 2007.
- [108] N. Bertschinger and T. Natschlager. *Neural Comput.*, 16:1413, 2004.
- [109] D. Sussillo and L. F. Abbott. *Neuron*, 63:544, 2009.
- [110] N. Packard. *Dynamic Patterns in Complex: Adaptation towards the edge of chaos*. World Scientific, 1988.
- [111] H. Jaeger. *Fraunhofer Institute for Autonomous Intelligent Systems, Tech. rep.*, 148, 2001.
- [112] J. Farmer. *Physica D*, 4:366, 1982.
- [113] A. Neyer and E. Voges. *IEEE J. Quantum Electron.*, QE-18:2009, 1982.
- [114] L. Larger, P.-A. Lacourt, S. Poinot, and M. Hanna. *Phys. Rev. Lett.*, 95:043903, 2005.
- [115] Y. Chembo Kouomou, P. Colet, L. Larger, and N. Gastaud. Chaotic breathers in delayed electro-optical systems. *Phys. Rev. Lett.*
- [116] K. E. Callan, L. Illing, Z. Gao, D. J. Gauthier, and E. Scholl. *Phys. Rev. Lett.*, 104:113901, 2010.
- [117] A. Argyris, D. Syvridis, L. Larger, V. Annovazzi-Lodi, P. Colet, I.O. Fischer, J. Garcia-Ojalvo, C. R. Mirasso, L. Pesquera, and K. Alan Shore. Chaos-based communications at high bit rates using commercial fibre-optic links. *Nature*, 438:343–346, 2005.

- [118] L. Larger. *Phil. Trans. Roy. Soc. A*, 371:20120464, 2013.
- [119] X. S. Yao and L. Maleki. *Electron. Lett.*, 30:1525, 1994.
- [120] Y. K. Chembo, K. Volyanskiy, L. Larger, E. Rubiola, and P Colet. *IEEE J. Quantum Electron.*, 45:178, 2009.
- [121] L. Maleki. *Nature Photonics*, 5:728, 2011.
- [122] X. Fang, B. Wetzell, J.-M. Merolla, J. M. Dudley, L. Larger, C. Guyeux, and J. M. Bahi. *IEEE Trans. on Circ. and Syst. I*, 61:888, 2014.
- [123] A. B. Cohen, B. Ravoori, T. E. Murphy, and R. Roy. *Phys. Rev. Lett.*, 101:154102, 2008.
- [124] T. E. Murphy et al. *Phil. Trans. R. Soc. A*, 368:343, 2010.
- [125] B. Ravoori, A. B. Cohen, J. Sun, A. E. Motter, T. E. Murphy, and R. Roy. *Phys. Rev. Lett.*, 107:034102, 2011.
- [126] L. Illing, G. Hoth, L. Shareshian, and C. May. *Phys. Rev. E*, 83:026107, 2011.
- [127] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer. *Optics Express*, 20:3241, 2012.
- [128] R. Martinenghi, S. Rybalko, M. Jacquot, Y. Chembo, and L. Larger. *Phys. Rev. Lett.*, 108:244101, 2012.
- [129] V.S. Udaltsov, L. Larger, J.-P. Goedgebuer, M. W. Lee, E. Genin, and W.T. Rhodes. *IEEE Trans. on Circ. and Syst.*, 49:1006, 2002.
- [130] M. Peil, M. Jacquot, Y. Kouomou, L. Larger, and T. Erneux. *Phys. Rev. E*, 79:026208, 2009.
- [131] L. Weicker, T. Erneux, O. D’Huys, J. Danckaert, M. Jacquot, Y. Chembo, and L. Larger. *Phil. Trans. R. Soc. A*, 319:20120459, 2013.
- [132] G. R. G. Chengui, A. F. Talla, J. H. Talla Mbe, A. Coillet, Khaldoun Saleh, L. Larger, P. Woafu, and Y. K. Chembo. *J. Opt. Soc. Am. B*, 31:2310, 2014.

-
- [133] J. H. Talla Mbe, A. F. Talla, G. R. Goune Chengui, A. Coillet, L. Larger, P. Wofo, and Y. K. Chembo. *Phys. Rev. E.*, 91:012902, 2015.
- [134] T. Matsumoto, L. O. Chua, and M. Komuro. The double scroll. *Int. J. Circuit Theory Appl.*, 14:117–146, 1986.
- [135] I. Kovacic and M. J. Brennan. *The Duffing Equation: Nonlinear Oscillators and Their Behaviour*. Wiley, 2011.
- [136] E. Ott. *Chaos in Dynamical Systems*. Cambridge University Press, 2002.
- [137] W. K. S. Tang, G. Q. Zhong, G. Chen, and K. F. Man. *IEEE Trans. Circuits Syst. Fundam. Theory Appl.*, 48:1369, 2011.
- [138] M. E. Yalcin and S. Ozoguz. *Chaos*, 17:033112, 2007.
- [139] B. A. Marquez, J. J. Suarez-Vargas, and Javier A. Ramirez. *Chaos*, 24:033123, 2014.
- [140] J. D. Meiss. *Differential Dynamical Systems*. SIAM, 2007.
- [141] D. Brunner, M.C. Soriano, C.R. Mirasso, and I. Fischer. *Nat. Comm.*, 4:1364, 2013.
- [142] F. Duport, A. Smerieri, A. Akrouf, M. Haelterman, and S. Massar. *Sci. Rep.*, 6:22381, 2016.

Abstract:

Artificial neural networks are systems prominently used in computation and investigations of biological neural systems. They provide state-of-the-art performance in challenging problems like the prediction of chaotic signals. Yet, the understanding of how neural networks actually solve problems like prediction remains vague; the black-box analogy is often employed. Merging nonlinear dynamical systems theory with machine learning, we develop a new concept which describes neural networks and prediction within the same framework. Taking profit of the obtained insight, we a-priori design a hybrid computer, which extends a neural network by an external memory. Furthermore, we identify mechanisms based on spatio-temporal synchronization with which random recurrent neural networks operated beyond their fixed point could reduce the negative impact of regular spontaneous dynamics on their computational performance. Finally, we build a recurrent delay network in an electro-optical setup inspired by the Ikeda system, which at first is investigated in a nonlinear dynamics framework. We then implement a neuromorphic processor dedicated to a prediction task.

Keywords: nonlinear, photonic, neuromorphic

Résumé :

Les réseaux de neurones artificiels constituent des systèmes alternatifs pour effectuer des calculs complexes, ainsi que pour contribuer à l'étude des systèmes neuronaux biologiques. Ils sont capables de résoudre des problèmes complexes, tel que la prédiction de signaux chaotiques, avec des performances à l'état de l'art. Cependant, la compréhension du fonctionnement des réseaux de neurones dans la résolution de problèmes comme la prédiction reste vague ; l'analogie avec une boîte-noire est souvent employée. En combinant la théorie des systèmes dynamiques non linéaires avec celle de l'apprentissage automatique (Machine Learning), nous avons développé un nouveau concept décrivant à la fois le fonctionnement des réseaux neuronaux ainsi que les mécanismes à l'œuvre dans leurs capacités de prédiction. Grâce à ce concept, nous avons pu imaginer un processeur neuronal hybride composé d'un réseaux de neurones et d'une mémoire externe. Nous avons également identifié les mécanismes basés sur la synchronisation spatio-temporelle avec lesquels des réseaux neuronaux aléatoires récurrents peuvent effectivement fonctionner, au-delà de leurs états de point fixe habituellement utilisés. Cette synchronisation a entre autre pour effet de réduire l'impact de la dynamique régulière spontanée sur la performance du système. Enfin, nous avons construit physiquement un réseau récurrent à retard dans un montage électro-optique basé sur le système dynamique d'Ikeda. Celui-ci a dans un premier temps été étudié dans le contexte de la dynamique non-linéaire afin d'en explorer certaines propriétés, puis nous l'avons utilisé pour implémenter un processeur neuromorphique dédié à la prédiction de signaux chaotiques.

Mots-clés : nonlinéaire, photonique, neuromorphiques

The logo for SPIM (École doctorale SPIM) features a stylized white 'S' on a green horizontal bar, followed by the letters 'PIM' in a large, white, sans-serif font.

■ École doctorale SPIM 1 rue Claude Goudimel F - 25030 Besançon cedex

■ tél. +33 (0)3 81 66 66 02 ■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

The logo for the University of Franche-Comté (UFC) consists of a large 'U' and 'FC' with a vertical bar between them, and the text 'UNIVERSITÉ DE FRANCHE-COMTÉ' below.