Vorhersage und Kontrolle chaotischer Systeme mithilfe Lernender Maschinen

Youssef Abdelhamid Mabrouk

Munich 2020

Predicting and Controlling Chaotic Systems with Machine learning

Youssef Abdelhamid Mabrouk

Munich 2020

Predicting and Controlling Chaotic Systems with Machine learning

Youssef Abdelhamid Mabrouk

Master thesis
at the faculty of physics
of the Ludwig-Maximilian University
of Munich

submitted by
Youssef Abdelhamid Mabrouk
from Tunis

Munich, September 10, 2020

Supervisor: PD Dr. Christoph Räth
Examination date: September 10, 2020

Contents

List of Figures

List of Tables

Abstract

The need for methods for the analysis of complex data in practically all modern scientific disciplines is clear. The need for methods for the analysis of complex dynamical data is equally clear because modern scientific measurement devices can resolve a remarkably wide range of temporal scales and thus cover a wide diversity of multi scale physics. Here, we investigate the difficulties encountered while detecting, predicting and controlling chaotic dynamical systems in cases where there is lack of knowledge about the underlying physical mechanisms. We implement a machine learning method called reservoir computing that extracts a dynamical rule from the data and compare the performance of this method to standard attractor reconstruction methods. The comparison is made on the basis of synthetic chaotic data defined from the numerical solution of chaotic ordinary differential equations. We show that while reservoir computers allow an automated and potentially efficient data processing scheme, they may result in many difficulties related with the choice of empirical fitting parameters.

Chapter 1

Introduction

Physical systems resolvable on a dynamical scale are found in a rapidly increasing number of situations due to the advance of modern scientific instruments[3, 4, 5, 6, 7]. For example, the spatio-temporal fluctuations of pressure waves inside a rocket combustion chamber can now readily be resolved with help of an optical measurement device[1]. Physical systems exhibiting chaotic behaviour deserve particular attention because of the time-efficient and low-energy control properties that chaotic systems may potentially have[8]. For example, feedback control of chaotic instabilities within the cardiac tissue allows to avoid massive electric shocks of an implanted ventricular defibrillator with non-invasive perturbations[9].

Given a measurement consisting of chaotic fluctuations and no knowledge about the underlying dynamics, the question is often whether it is possible to detect transitions between qualitatively different dynamical regimes in *real-time* and how to prescribe a weak external force that constrains the system to evolve in the dynamical regime of interest in case such a transition is detected. Having such a knowledge at hand may be of advantage when it comes to the need for suppressing chaotic fluctuations by rendering the dynamics periodic [8]. Of notable relevance is also the need of sustaining chaotic fluctuations in case the system has entered an unwanted transiently chaotic regime[10]. While the indicated cases have been subject to extensive previous investigation, and several experimentally proven methods have been established, common to all established methods is the need for attractor reconstruction routines that can often be not practical[8, 10].

More generally, attractor reconstruction and identification of chaotic data occurs to be severely difficult when it comes to analysing data measured from a physical environment and when little is known about the physics of the concerned environment[11]. This can be made visible by noticing that some of the simplest examples of chaotic systems described by one-dimensional maps admit a temporal correlation equivalent to that of random noise[12]. This means that on average knowledge of the value of the state variable at a specific point in time provides strictly no knowledge about the value of the state variable at a later time point. One consequence of this is the inefficiency of statistical methods that rely on linear moments of the data to detect and extract temporal patterns from the data. Beside this a substantial problem is that one is often confronted with noisy and partial measurements of the physical variable under consideration.

With the emergence of statistical learning methods it is of interest to investigate how these can be used in order to alleviate the above indicated efforts involved with attractor reconstruction. Among the learning methods widely employed for the analysis of temporal data, a particularly appealing class is that of reservoir computers[13]. This class is distinguished by the introduction of an internal state variable that exhibits a dynamical behaviour driven by the observational data. One advantage of such a method is that the internal state variable can be updated incrementally in step with the incremental measurements of the dynamical system under observation. An other advantage is the absence of any need of preprocessing the data before it is fed into the internal states.

Relying on statistical learning methods has the severe disadvantage of leading to hardly interpretable results. For example, in the analysis of temporal data with reservoir computers interpretability would consist of a mechanism describing the dynamic interplay between the driving observational data and the driven internal states. From such a mechanism one may ideally deduce a clear description of how to define the reservoir architecture in order to achieve optimal performance. Despite the considerable amount of insight that is laboured from statistical learning theory and which is fruitfully applied in common practice, it is often not immediately clear how to apply such an insight to realistic situations. For example, in order to calculate an estimation for the memory or the generalizability of a particular model, statistical learning theory makes the assumption of independent identically distributed random data[14]. This assumption disregards any kind of temporal behaviour and thus motivates investigating the question of how one can take the temporal behaviour of the data into account when analysing performance. The investigation of this question is to large extent the core problem treated in this work.

In the first chapter we introduce briefly the main properties of chaotic systems on hand of simple examples. In the first part of the second chapter, we first shortly discuss the main established data analytical methods of attractor reconstruction and modelling of chaotic data. The second part is devoted to the method of reservoir computers and how these can be used for the data analysis of chaotic data.

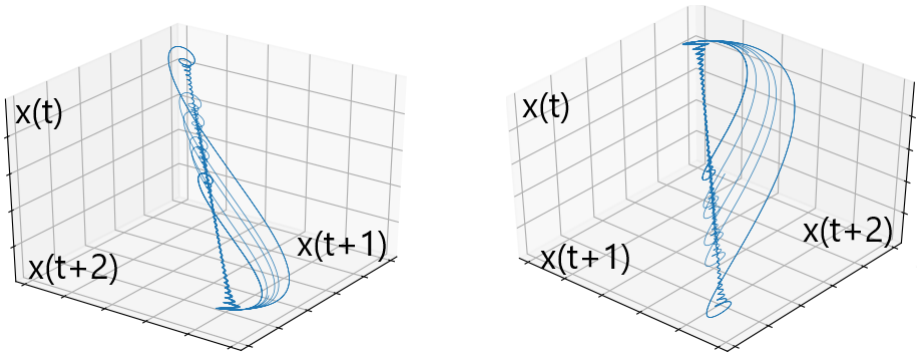

Figure 1.1: Example of chaotic attractor reconstructed from combustion noise measured from a rocket combustion chamber[1]. The shown reconstruction is obtained from a scalar time series $\mathbf{x}_t$ consisting of pressure fluctuations resolved in a millisecond scale.

Chapter 2

Theoretical foundations

2.1 Coupled dynamical systems

As a starting point we consider the simplified case of two separate linear dynamical variables x_1 and x_2 defined on the real axis, as follows

$$\dot{x_1} = \alpha x_1, \qquad \dot{x_2} = \alpha x_2 \tag{2.1}$$

where we have associated with each of these dynamical systems the positive growth rate α. It is interesting to ask how the exponential growth of these two variables changes if we introduce a coupling term to the dynamics, such that the evolution of each variable is determined by its intrinsic growth α and the coupling to the other variable. In the simplest form, we may define such a coupling as

$$\dot{x_1} = \alpha x_1 + \varepsilon \cdot (x_2 - x_1), \qquad \dot{x_2} = \alpha x_2 + \varepsilon \cdot (x_1 - x_2) \tag{2.2}$$

Here, ε denotes the strength of the coupling force added to the dynamics. The limiting case of a coupling magnitude negligible compared with the growth rate corresponds the above define situation of two exponentially growing separate variables. The limiting case of a growth rate negligible compared with the coupling strength corresponds to the case where an initial difference between the two variables exponentially decays to zero with a rate equal to two. It is interesting to ask how the transition between these two extremes takes place if we gradually change the strength of the coupling term ε. This gradual change may be illustrated by considering the dynamics of the joint state (x_1, x_2) in the two dimensional phase space near the subset consisting of the identity line. If the coupling is zero, then this subspace is invariant because if x_1 and x_2 are initiated at the same point they stay the same. But this subspace is unstable because if x_1 and x_2 are initiated in similar but slightly different points then the difference will grow linearly. By introducing coupling we might have a chance to change the stability of this subspace because the coupling term acts as a kind of feedback that readjusts each variable with respect to the difference between the two variables. We want to know how small ε is allowed to be with the constraint that

the invariant subset ($x_1 = x_2$) is stable. In order to do this we write the above system of coupled equations in matrix notation as follows,

$$\begin{matrix} \dot{x_1} \\ \dot{x_2} \end{matrix} = \begin{matrix} a-\varepsilon & \varepsilon \\ \varepsilon & a-\varepsilon \end{matrix} \cdot \begin{matrix} x_1 \\ x_2 \end{matrix} \tag{2.3}$$

The characteristic equation of the above defined system is $(a - \varepsilon - \lambda)^2 = \varepsilon^2$. The solution of this characteristic equation is the eigenvalue $\lambda = a - 2\varepsilon$ which is associated with the eigenvector corresponding to the direction perpendicular to the identity line. It follows that $a \leq 2\varepsilon$ is the condition of stability of the invariant set ($x_1 = x_2$). This asymptotic convergence of two dynamical variables to a state where these are related by a static functional relation is referred to as synchronization. In our case, the functional relation consists of the identity, but there are situation where this is not the case. Awareness of this kind of mechanisms can be of notable importance. For example, in a time-series analytical framework, this kind of mechanism can be used to gradually adjust model parameters such that the model state variable converges to the observational data[15]. In fact, the indicated reference is related to the case where the equations of motion underlying the dynamical observational data are known and the specific values of the parameters defining the equations of motion are lacking. In that case, stable synchronization to the correct parameter values could be ensured since the full set of equations is available. It is interesting to ask whether this kind of approach of extraction of dynamical rules from observed data could be applied in cases where there is strictly no knowledge about the underlying dynamics.

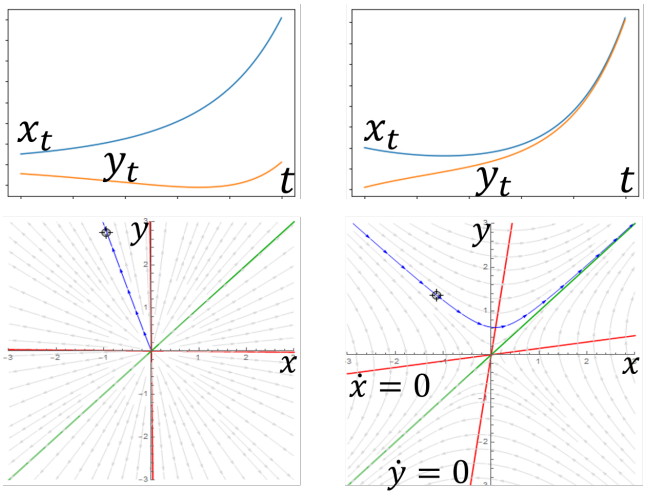

Figure 2.1: Example of synchronization of two coupled one dimensional linear dynamical systems. The left panels indicate the time resolution and phase plane for a coupling below threshold. The right panels indicate the time resolution and phase plane for a coupling above threshold.

2.2 Chaotic dynamical systems

We consider the evolution of the one dimensional discrete dynamical system given by

$$x_{t+1} = \begin{array}{c} 2x_t : x_t < \frac{1}{2} \\ 2x_t - 2 : x_t > \frac{1}{2} \end{array} \tag{2.4}$$

In contrast to the dynamical system considered in the previous section, whose discrete correspondent can be simply identified with the linear one dimensional map $x_{t+1} = \alpha x_t$, the map defined by equation 2.4 exhibits a similar linear growth when the dynamical variable belongs to the first half of the unit interval, but the growth is abruptly interrupted by a discontinuous nonlinearity that comes into play when the dynamical variable enters the second half of the unit interval. From this dynamic interplay between linear instability near the unstable fixed point $x = 0$ and nonlinear saturation above the threshold $x = \frac{1}{2}$ we expect the onset of some kind of sustained oscillation. To clarify the kind of this oscillation we consider the way how the binary expansion of the value of x changes by application of the map. For a number in the unit interval, the first coefficient in its binary expansion is zero. Multiplying such a number by two is shifting the coefficients of its binary expansion. If the multiplication results in a first coefficient that is not zero, that is, if the number belongs to the second half of the unit interval, then the definition of the map resets the first coefficient to zero. Thus the action of this map is equivalent to increasing the order of the binary coefficients by one order while setting the first coefficient to zero. From this observation we can make some qualitative statements about the dynamics. The increase of order of the coefficients indicates that the smallest details contained within the lowest orders of the expansion of some initial condition x_0 will become important at some point of the evolution. We refer to this property as sensitivity to initial conditions. This is a short-term property of the dynamical system, because we now know that two points initiated near each other will be separated from each other after some finite amount of time. But we can also make statements about the long-term properties of this dynamical system. If the considered initial value x_0 is a rational number then the resulting trajectory is periodic. This is because increasing the order of the binary coefficients of a rational number results in a periodic sequence of numbers with period p equal to the period of that rational number. Similarly, If the considered initial value x_0 is an irrational number then the resulting orbit is not periodic. Because the rational numbers have measure zero, it seems that the typical choice of the initial condition results in a dynamical behaviour that is not periodic. Because the rational numbers are dense within the real numbers, it seems that arbitrarily close to a given point of the nonperiodic orbit one can find a point that belongs to a periodic orbit. To make more precise statements about the dynamics, it is necessary to consider the stability of the indicated periodic orbits. The stability of a fixed point can be assessed by calculating the value taken by the Jacobian at that fixed point. If this value is superior than one, the fixed point is unstable. This is because the Jacobian corresponds to the multiplicative growth rate of a small disturbance of the fixed point. Similarly, for a periodic orbit consisting of p points stability can be deduced by calculating the average value of the Jacobian evaluated at the points that constitute the

periodic orbit. The logarithm of the average value of the Jacobian is referred to as the Lyapunov exponent. If the dynamical system has more than one degree of freedom, which is not the case here, we similarly define the Lyapunov spectrum to be the logarithms of the average values of the eigenvalues of the Jacobian matrix. For a system that has more than one dimension a spectrum of exponents is needed because stability may change depending on the direction in which the disturbance from the periodic point is made. We apply the former definition for the case of the Bernoulli map as follows,

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \ln \sum_{i=1}^{n} f\prime(x_i) = \ln 2 \tag{2.5}$$

The above estimate for the Lyapunov exponent is the limiting value of an infinitely long sequence. In a computer simulation, we would approach that value as we increase the number of steps defining the sequence. Because we have to stop the computer simulation at some point computer estimates depend on the initial value where we start generating our sequence. If we accidentally start our orbit very close to a particular unstable periodic orbit, it is clear that repulsion from that unstable orbit will need some time and our short-time estimate of the Lyapunov exponent will be similar to that of the unstable periodic orbit. Because the set of unstable periodic orbits is countably infinite, it is clear that our short-term estimate will depend on the initial point where we start. Like the Lyapunov exponent, other properties of the sequence will also depend on the initial value where we start our simulation. Because the synchronization threshold of coupled dynamical systems depends on the properties of the separate dynamical systems, one consequence of the short-term variability is the dependence of synchronization thresholds on the initial value for systems that exhibit similar properties to the map presented here.

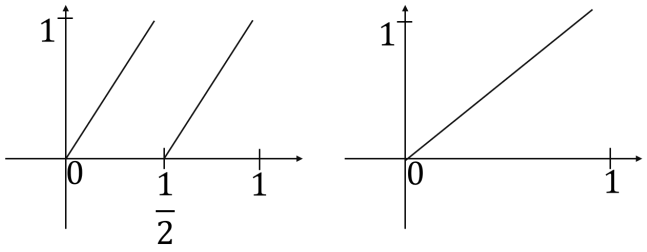

Figure 2.2: Example of a nonlinear dynamical system contrasted with a linear dynamical system. The state space of the dynamical variable is the unit interval. The map defining the dynamics corresponds to the two lines that increase from 0 to 1 and from $\frac{1}{2}$ to 1 both with a slope equal to 2

Chapter 3

Computational methods

We now consider the situation where we are given a sequence of measurements $\{\mathbf{x}_t\}$ with $t \in [0, T]$ of some physical variable $\mathbf{x}_t \in R$. At a first step we wish to investigate the question of whether there is a deterministic dynamical system of the form

$$\mathbf{z}_{t+1} = f(\mathbf{z}_t) \tag{3.1}$$

underlying the sequence of measurements such that our experimentally accessible variable $\mathbf{x}_t$ is a function $\phi : R^N \to R$ of the dynamical variable $\mathbf{z}_t \in R^N$ according to $\mathbf{x}_t = \phi(\mathbf{z}_t)$. Answering this question is important because it indicates whether the system is predictable or not and most importantly provides an estimation for the time-scales on which predictability may be possible. The main challenge is that the typical and most often encountered situation consists of a measurement variable $\mathbf{x}_t$ that is only a partial measurement of the dynamical variable $\mathbf{z}_t$. In the simplest case $\mathbf{x}_t$ would be simply a projection of $\mathbf{z}_t$ such that only a few components of $\mathbf{z}_t$ are accessible. This can be problematic because the existence of a map $f : \mathbf{z}_t \to \mathbf{z}_{t+1}$ does not necessarily imply the existence of a map $g : \mathbf{x}_t \to \mathbf{x}_{t+1}$ associated with the observed variable. For example, knowledge of the position of a harmonically oscillating spring mass at a specific time is not sufficient to predict the position of the mass one time step later. For this one needs to know whether the mass is moving downwards or upwards, that is, one needs the full coordinates in the two dimensional phase plane rather than only one of them. Thus, the transition function $g : \mathbf{x}_t \to \mathbf{x}_{t+1}$ is in this case clearly not well-defined. This kind of situation is most typically treated with help of delay coordinate methods. This kind of method consists of mapping the n-dimensional observational data to an N-dimensional point cloud that conserves the neighbourhood information of the attractor underlying the data. By conservation of neighbourhood information we mean that neighbour points in the attractor are mapped to neighbour points in the constructed N-dimensional space. If such a construction is found, one could for example approximate the future $\mathbf{x}_{t+1}$ of given point $\mathbf{x}_t$ by the future of its neighbour in the reconstructed phase space. A difficulty of the method of delay coordinates is that the attractor dimension where the dynamics is taking place is in general not known and has to be in some way estimated. A further problem is that the number of needed data points is exponential in the dimension of attractor.

There has been a number of methods developed in order to avoid the above described reconstruction effort. For example, the fact that it is not possible to define a transition function $g : \mathbf{x}_t \to \mathbf{x}_{t+1}$ does not prevent us from looking for a transition function according to $g : \mathbf{y}_t = (\mathbf{x}_{t-d}, \mathbf{x}_{t-d+1}, .., \mathbf{x}_{t-1}) \to \mathbf{x}_{t+1}$ that maps the history of our observable variable to its value at one future time point. Most conveniently, this may be done by defining a regression $\varepsilon =< g(\mathbf{x}_{t-d}, \mathbf{x}_{t-N+1}, .., \mathbf{x}_{t-1}) - \mathbf{x}_t >_T$ with respect to g. In principal, choosing the dimension d sufficiently large ensures all the contributing terms from the history to be taken into account. Thus, this alleviates the effort of attractor reconstruction by letting the regression procedure decide which terms in the history determine the future.

The direct way of inference would be to approximate the function $g : R^d \to R$ by a linear combination of suitably chosen functional basis $g(\mathbf{y}_t) = \sum c_i p_i(\mathbf{y}_t)$. The choice of the number of basis functions p_i and the functional basis would depend on the knowledge available for the system under consideration. For example, the functions p_i could be the leading order polynomial functions. In such a setting, the regression is reduced to determining the set of coefficients c_i that minimize the regression error. One disadvantage of this procedure is that it requires some minimum knowledge about the dynamics under consideration in order to choose the basis functions. An other problem is that the number of coefficients grows exponentially in the delay dimension that we need to choose. This makes this method unfavourable for treating spatio-temporal systems.

A closely related but slightly different alternative is to introduce a dynamical variable $\mathbf{y}_t$ that is conjugate to the observational data. The idea behind this is that the internal state $\mathbf{y}_t$ should encode the information that would be encoded in a reconstructed state of $\mathbf{x}_t$. In this case the dynamics of the internal variable is determined by

$$\mathbf{y}_{t+1} = g(\mathbf{x}_t, \mathbf{y}_t) \tag{3.2}$$

While this class of methods is suitable for real-time assimilation and forecasting of sequential data, the above discussed delay coordinate method can be more suitable for modelling and interpreting the dynamics.

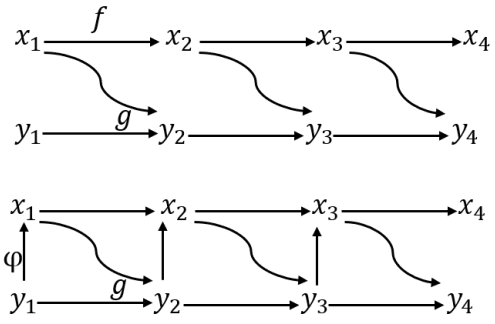

Figure 3.1: Internal state representation. The observed variable $\mathbf{x}_t$ evolves autonomously as a projection of $\mathbf{z}_t$ which evolves by f. The internal variable $\mathbf{y}_t$ is first driven by $\mathbf{x}_t$ in the learning phase and then becomes autonomous in the prediction phase.

3.1 Attractor reconstruction

The delay coordinates are defined based on the sequence of data $\{\mathbf{x}_t\}$ according to $\{\mathbf{y}_t\} = \{(\mathbf{x}_t, \mathbf{x}_{t+\tau}, \mathbf{x}_{t+2\tau}.., \mathbf{x}_{t+d\tau})\}$. In this definition, d signifies the number of delays used to construct each vector $\mathbf{y}_t$ and τ signifies the delay variable. The number of delays is equivalent to the dimension of the reconstructed attractor. In principal, the dimension d must be at least twice as large as the intrinsic attractor dimension in order to ensure conservation of neighbourhood information in the attractor defined by the delay coordinates. On the other hand, the delay variable τ is needed to take into consideration the effect of oversampling or noisy data. This can be illustrated as follows. In the example of the harmonically oscillating spring mass, a partial measurement would be the series of positions $\{\mathbf{x}_t\}$ of the mass. But the trajectory in the two dimensional phase space is an ellipse. In order to reconstruct the trajectory from the scalar measurement series, the delay coordinates in this case would be simply $\{\mathbf{y}_t\} = \{(\mathbf{x}_t, \mathbf{x}_{t+\tau})\}$. This would give us a two dimensional point cloud. The points which are seen as neighbours in the scalar time series but are not neighbours in the two dimensional trajectory consist of the points that have the same position but are moving in different directions. It is clear that these points will be separated in the delay coordinate reconstruction, since the delayed value of the scalar time series will depend on the direction on which the mass is moving. Thus, it is intuitively clear that the delay coordinate construction recovers the neighbourhood information from the scalar time-series. The tunable delay variable τ is needed because if one chooses the delay much smaller than the intrinsic time scale of the system, the delayed values would be simply the same since the dynamics has not considerably evolved. In our simplified example, this would correspond to the identity line. On the other hand, if the delay is chosen too large the delayed values may become decorrelated and the delay reconstruction would simply look like noise. This is not illustrated by our simplified example, since the loss of correlation is typically implied by the nonlinear dynamics.

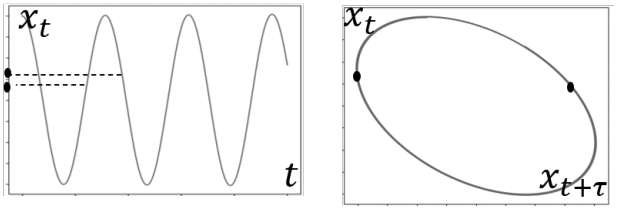

Figure 3.2: Delay coordinates. The left panel indicates the time evolution of the position variable of harmonic oscillator. The right panel indicated the reconstruction obtained from the scalar time series via delay coordinates.

3.2 Attractor prediction

Once we have obtained a reconstruction $\{\mathbf{y}_t\}$ that is equivalent to the attractor under consideration, we can make use of several methods in order to determine the dynamical map $g : \mathbf{y}_t \to \mathbf{y}_{t+1}$. As mentioned earlier, the simplest way is expand the map g as a linear combination of polynomial basis functions. In the following, we illustrate this on hand of the three dimensional Lorenz chaotic flow. For simplicity, we will take the data $\{\mathbf{y}_t\}$ to be the three dimension vector field, rather than a reconstruction as it would be in a real case. We denote the three dimensional temporal data as $\mathbf{y} = (y_1, y_2, y_3)$. We proceed by writing the dynamical map as $g_i(\mathbf{y}) = \sum \mathbf{c}_i p_i(\mathbf{y})$. We have restricted the polynomial functions to second order because this is the highest order appearing in the Lorenz flow. The first index of the coefficients c_{ij} can take three values corresponding to the three components of the vector flow. The second index of the coefficients c_{ij} corresponds to the different orders for each component, and takes three times three values since we have considered only the three leading order terms. Now the goal is to find an optimal set of coefficients c_{ij} such that each vector $\mathbf{y}_t$ from the time-series is matched to its consecutive vector $\mathbf{y}_{t+1}$ by the map g. Please note that the index t signifies the sequential index within the time series while i is used for the components. To proceed, we write the regression error as $\varepsilon =< \parallel g(\mathbf{y}_t) - \mathbf{y}_{t+1} \parallel^2 >_T$, where $\parallel \cdot \parallel$ signifies the L_2 norm and the $< \cdot >_T$ indicates the time average over the all the points contained within the sequence. The fact that the dynamical map is a linear combination of the polynomial functions allows us to write the regression problem as a linear system for the coefficients. This is convenient because the optimal solution of coefficients can be written in closed form. To make this visible, we write the derivative of the regression error with respect to the coefficients as follows

$$\partial_{c_{ij}}\varepsilon =< \partial_{c_{ij}} \parallel g(\mathbf{y}_t) - \mathbf{y}_{t+1} \parallel^2 >_T =< \partial_{\mathbf{c}_i} \sum (\mathbf{c}_i p_i(\mathbf{y}_t) - \mathbf{y}_{t+1})_j^2 >_T = 0 \tag{3.3}$$

thus

$$c =< g(\mathbf{y}_t)^T g(\mathbf{y}_t) >_T^{-1} < g(\mathbf{y}_t)\mathbf{y}_{t+1} >_T \tag{3.4}$$

Thus, in order to obtain the optimal set of coefficients we need to compute the image of $\mathbf{y}_t$ with respect to g for each data point in the time-series we have. Then the coefficients can be extracted from the correlation matrix of $\mathbf{y}_{t+1}$ and $(\mathbf{y}_t)$ and the inverse of the covariance matrix of $(\mathbf{y}_t)$. We note that it is not always the case that the covariance matrix is invertible. In case it is not invertible, the standard way to proceed is to modify the regression error such that the inversion is done with respect to the regularized covariance matrix. Adding the regularization term in the regression has also the advantage that it makes the matrix inversion numerically more stable, since it is equivalent to adding a positive number to the diagonal elements of the covariance matrix.

An example of the above described procedure is shown below. The data has been generated by numerically integrating the Lorenz chaotic system with help of fourth order Runge-Kutta method. The details of the step-size parameter and the numerical implementation can be found in the appendix. The first panel in Figure 3.3 shows a comparison between the trajectory generated from the learned model and the correct trajectory. While

it is difficult to distinguish between the two trajectories when visualized in R^3, a closer look at the projection along one axis shows that the two trajectories are indistinguishable only for a short time, than they become quantitatively different although the qualitative behaviour is the same. It is worth mentioning that this separation of trajectories is not evident. The fact that the trajectory is chaotic and that it has one positive Lyapunov exponent tells us that two slightly different initial conditions will maintain correlation only up to one Lyapunov exponents if they are evolved under the same dynamics. We have considered two trajectories starting from the same initial condition but evolving under slightly different dynamics. The second panel in Figure 3.3 shows the regression error resolved with respect to time, that is, $\varepsilon_t = \| g(\mathbf{y}_t) - \mathbf{y}_{t+1} \|$. It is seen that this temporal evolution is marked by the appearance of irregular peaks interrupting the low values taken by the regression error. Taking this kind of outliers into consideration is of importance, because eliminating the points that can not be fitted can often considerably improve the regression quality. We note also that there is a kind of repeating structure that appears just before the peaks in the regression error. This is worth noticing, because the main assumption of our regression analysis was that the the regression errors are independent identically distributed random variables, and the appearance of correlations indicates that this assumption may be improved. The temporal evolution of the error will be treated in more detail later in this work in a slightly different setting, namely in reservoir computers, which is the main focus of this work.

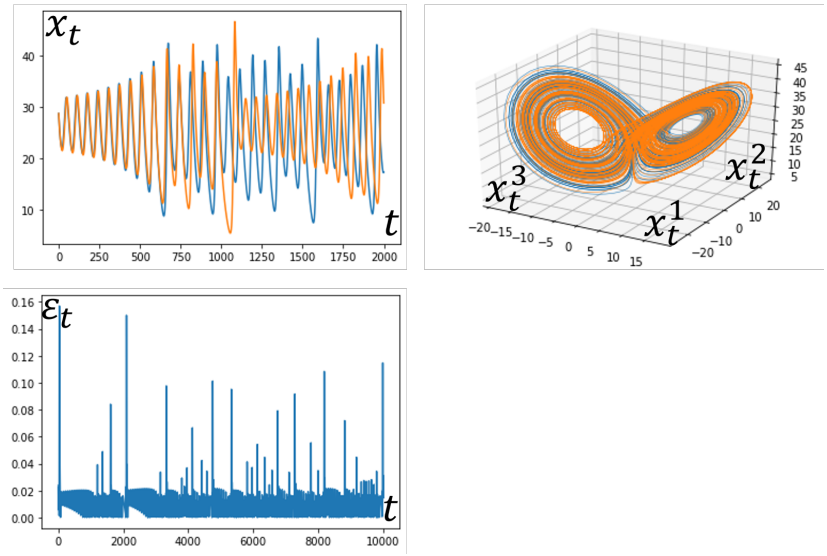

Figure 3.3: Equations of motion from data via polynomial basis expansion. The first two panels show a superposition of the trajectory obtained from the learned model and the trajectory obtained from the defining equations of motion. The third panel illustrates the time resolution of the regression error.

3.3 Attractor control

One we have extracted an effective model from the data we are ready to analyse questions related with control of the chaotic system under consideration. In this context, control refers to the situation where we need to change the dynamical regime of the chaotic system by applying a small size external force or by slightly changing a control parameter. For example, this could be the situation where the system has entered a chaotic regime as a result of an abrupt parameter change and we want to force the system to the periodic state without having to change the parameter considerably or to exert considerable forces on the system. The reason why this may be possible can be illustrated with help of the simple example of the quadratic map. As the control parameter of the quadratic map is varied the dynamics changes qualitatively such that at each change an attracting cycle becomes non attracting and an attracting cycle with a longer period is developed. It becomes clear that repeating this scenario many times will result in a large number of non-attracting cycles in phase space. Thus, even for a parameter value where the system is chaotic, if the state variable is initialized in one of the points of a cyclic trajectory, it will remain in that cyclic trajectory. Conversely, if the trajectory is not initialized in a cycle, we can try to force it to stay on the cycle that we choose.

One particularly simple way to control the chaotic trajectory is to apply an external force to the system such that the force is proportional to the difference between the chaotic trajectory and the periodic trajectory of interest,

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t) + \varepsilon(\mathbf{x}_t^p - \mathbf{x}_t) \tag{3.5}$$

where $\mathbf{x}_t$ denotes the system evolution within the chaotic regime f under the application of the control force, ε denotes a scaling parameter that manipulates the magnitude of the control force, and $\mathbf{x}_t^p$ denotes the periodic trajectory that wish to achieve. Ideally the parameter ε has to be as small as possible, otherwise this would contradict the main motivation of the control procedure. It is clear that whenever the trajectory approaches the cycle the control force will be equal to zero. Because the unstable periodic orbits form a dense set within phase space, we do not expect the control force to be large in magnitude.

Thus, the main ingredient to perform the above described continuous time control procedure is the knowledge of the periodic trajectory $\mathbf{x}_t^p$ that we need the system to evolve in. Sine there has been a number of methods developed to extract the unstable periodic orbits of a chaotic system from its defining equation, this is what makes the extraction of the equations of motion from the data important for controlling chaotic systems. One particularly simple method to find the periodic trajectories of a chaotic flow is the delay feedback method. This consists in adding a feedback control force that is equal to the difference between the value of a trajectory and its delayed value,

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t) + \varepsilon(\mathbf{x}_{t-\tau} - \mathbf{x}_t) \tag{3.6}$$

The idea is that if the delay is chosen properly such that it is equal to the period of a periodic trajectory embedded within the chaotic attractor, then the control force should

vanish. Strictly speaking, this method allows to have only an estimation for the available periods, since the trajectory can be stabilized on a cycle that is not proper to the system. This means, if the delay variable is close but slightly different from one the period of an unstable periodic orbit, then the stabilized trajectory will have a periodic motion with a correspondingly close period. The behaviour of the controlled trajectory under variations of the control parameters is illustrated below for the example of the Lorenz system. In Figure 3.4 is shown the dependence of the average magnitude of the control force on the delay variable and the coupling constant ε. The average magnitude of the control force can be interpreted as a control error, since it would take large values if the orbit has not converged to a periodic orbit and it would be small otherwise. From Figure 3.4 we see that the values of the delay τ where the error ε admits minima correspond roughly to the minima in the (negative) temporal correlation function of the chaotic trajectory. This is important, because it allows us to interpret the maxima of the wide band spectrum of the Lorenz system as the periods of the unstable periodic orbits embedded within the attractor.

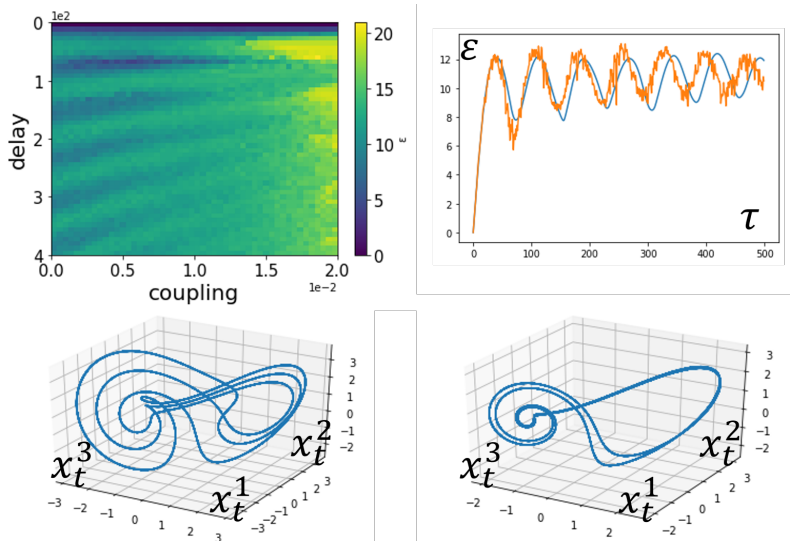

Figure 3.4: Extracting periodic trajectories. The first panel shows variations of the control error as a function of the coupling strength and delay variable. The second panel illustrates a the variation of the control error as a function of the delay variable for fixed coupling. The smooth curve corresponds to the case of zero coupling, and the fluctuating curve corresponds to nonzero coupling. The control error for zero coupling is closely related to the autocorrelation function(it is the same if the signal were scalar). The third and fourth panels show examples of periodic orbits extracted with this procedure.

In a similar way to the procedure descried by equation 3.5, where we apply a feedback force in order to render the trajectory periodic, and where the feedback force corresponds to the mismatch to the delayed trajectory, one can also consider the question of how to render the periodic chaotic in the case the system is in a transiently chaotic state, where it initially oscillates chaotically but after a finite time converges to a fixed point. One possible approach would be to define the control force in the form of a feedback but with respect to the mismatch between the observed trajectory and the chaotic trajectory, as follows

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t) + \varepsilon(\mathbf{x}_t^c - \mathbf{x}_t) \tag{3.7}$$

where $\mathbf{x}_t$ denotes the system evolution within the transiently chaotic regime f, and $\mathbf{x}_t^c$ now denotes the chaotic trajectory that we wish to make the system evolve in. Thus, the main requirement for this procedure consists of finding a dynamic rule that emulates the chaotic trajectory $\mathbf{x}_t^c$ in order to exert the feedback. As in the previous case, the question of importance is how small the feedback strength is allowed to be in order to sustain the chaotic oscillations. In a similar way to the previous paragraph, a suitable way to answer this question is to inspect the time average magnitude of the feedback force. If this average magnitude is large, then this is an indication that the sustainment is not functioning. The change of the outcome of this procedure as the control parameter ε changes is shown in Figure 3.5. Each pair of panels shows the comparison between the desired chaotic trajectory and the transient trajectory (blue and orange curves) for a specific value of the control strength. It is seen that for very small values $\varepsilon \sim 10^{-3}$ the transient trajectory can not be sustained by the feedback force and oscillates transiently and converges to a fixed point. At larger strengths the transient system maintains oscillations but one see clear distortions that distinguish the chaotic and the transient chaotic trajectories. At values approximately in the order of $\varepsilon \sim 10^{-1}$ the two trajectories become undistinguishable. We note that the transition occurs as a smooth transition as a function of ε, which can be seen from the smooth decease of the average control error as a function of ε.

In the previous paragraph, we have considered the two cases where the dynamical system under observation evolves either in a transiently chaotic or periodic regime and that one wants to render these regimes either to chaotic or periodic evolution. We have succeeded in doing this by applying a feedback force defined as mismatch to the desired trajectory. The main assumption of the setting we have defined was that the dynamic regime of the system does not change in time. Although this may be possible in some particular cases, it is more realistic to consider the situation where the system intrinsically changes between different regimes as time changes. In the simplest case, this may be due to fluctuations in the control parameters of the system under consideration.In principal, the method presented above can be directly applied to the case where the parameters change in time. But in the case that system parameters change in way that the system enters the desired state it would be convenient to switch the control off in order to make less effort in total. So the interesting question is whether one dynamically detect this kind of transitions and switch the control off or on depending on the current system state. Our approach is as follows. Assuming we have observed the system evolving in a particular

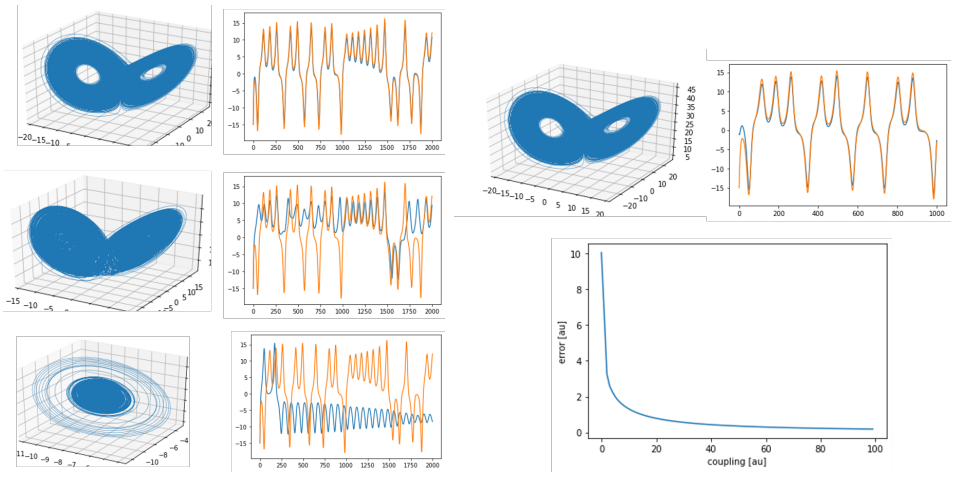

Figure 3.5: Sustaining the transient Lorenz chaotic system by feedback control. Each pair of figures (three dimension plot and temporal evolution) relates to a specific value of the control strength. The last plot shows the change of the time average of the control error in dependence no the control strength.

regime then we can learn its dynamical evolution, and by measuring the system states and feeding them in our learned model we can make prediction of these system states. If our predictions are reliable then a deviation from the prediction would immediately indicate a change of dynamical regime, and that would signal the need for switching on the control force. In turn, if the system is non being controlled on a specific regime and abruptly undergoes a transition to the regime of interest, this would be detectable from a decrease in the instantaneous value of the control error. Roughly, this means that two trajectories converge to each other under a feedback control easiest when their intrinsic dynamics are the same. The described procedure is illustrated in Figure 3.6. Shown are three different runs for cases where the bifurcation parameter ρ of the Lorenz system changes at random events in time between transiently chaotic and chaotic. The activation and deactivation of the control force is performed as described above. It is seen that our conditions are able to detect the transitions rather well. The three shown panels relate to different probabilities of occurrence of the transition. In each panel, the first plot shows the evolution of the trajectory, and below is shown the evolution of the control error, which clearly indicates the time points where the transition occurs.

3.4 Reservoir computers

The central idea underlying reservoir computers is that, under suitable conditions, coupled dynamical systems may achieve an asymptotic state of synchronization where the coupled systems are related via a static functional relation. More precisely, for the specific case of

a driving signal $\mathbf{x}_t$ coupled to a driven variable $\mathbf{y}_t$ according to

$$\mathbf{y}_{t+1} = g(\mathbf{x}_t, \mathbf{y}_t) \tag{3.8}$$

one can find sufficient conditions for the driven dynamics g that imply the existence of a function φ such that one has $\varphi(\mathbf{y}_t) = \mathbf{x}_t$ after sufficiently long time intervals. Under such conditions, inserting the measured sequence of data $\mathbf{x}_t$ into equation 3.8 allows us to compute the trajectory of the conjugate variable $\mathbf{y}_t$ during the measurement time interval T. This allows us to estimate the correlation matrix between the two variables $\mathbf{x}_t$ and $\mathbf{y}_t$, which we then use to systematically infer $\mathbf{x}_t$ from $\mathbf{y}_t$. Thus we have

$$\varphi = < \mathbf{y}_t^T \mathbf{y}_t >_T^{-1} < \mathbf{y}_t \mathbf{x}_t >_T \tag{3.9}$$

The above standing expression for the variable change φ is an optimal linear estimator. This reduces the training task to performing a single matrix inversion of the covariance matrix. Thus, the internal state variable $\mathbf{y}_t$ can be autonomously, and the dynamical variable of interest can be inferred from it through φ. Since our Ansatz for φ is approximate, during the autonomous dynamical phase the internal state variable is driven by a signal that is only approximately true. An exact prediction is possible only during a short time interval, after which only statistical properties of $\mathbf{x}_t$ may be concisely predicted. Thus arises the question of how to choose the driven dynamics g in order to obtain high quality predictions.

The common and widely used form of the driven dynamics is written as

$$g(\mathbf{x}, \mathbf{y}) = \tanh(W_{in} \cdot \mathbf{x} + W \cdot \mathbf{y}) \tag{3.10}$$

The resulting prediction scheme can compactly written in the form

$$\mathbf{y}_{t+1} = \tanh(W_{in} \cdot \mathbf{x}_t + W \cdot \mathbf{y}_t), \qquad \mathbf{x}_t = \varphi(\mathbf{y}_t) \tag{3.11}$$

Thus, this can be identified as a special kind of recurrent neural networks, where only the outermost layer φ is trained. The function approximation properties of neural networks are generically known to improve as the number of hidden is increased. In our case, this corresponds to choosing the internal states dimension orders of magnitude larger than the dimension of the driving signal. Since the time complexity of matrix inversion operations is $O(n^3)$, this would render the estimation in equation 3.9 computationally ineffective for large signals. The key idea that makes reservoir computers applicable to large multivariate data is the locality of the underlying dynamics. This means that future of each point at a specific time is only determined by the its neighbours at that specific time. For physical applications this is implied by the fact that physical interactions are often short ranged.

The method of reservoir computers as presented above seems to be rather straightforward, and it is a priori not clear which difficulties one may encounter in practice. In order to illustrate some of the main issues, we consider again the example of extracting an effective dynamical rule for the Lorenz chaotic system from data, but this time with reservoir computers. To start, we again generate the data $\mathbf{x}_t$ by integrating the chaotic Lorenz system and use the three dimensional vector as input data to the reservoir. We define the

random weights W via the random Erdös-Reny algorithm with edge creation probability $p = 0.02$ and set the number of nodes to $N = 500$. This choice for the generation of the weights is widely used in literature, but from our point of view, the main property of the weights is that they are sparse. An illustrative example of how the structure of internal weights W can be designed without randomization is the delay line network. Before we continue with our example, we briefly describe the construction of the delay line network for the case of a scalar signal $\mathbf{x}_t$. The network is defined by generating the weights W in a way such that only the elements directly below the diagonal are different from zero $W_{ii-1} \neq 0$ and the input weights constitute a vector that has the dimension of the internal states and where only the first component is different from zero. If the activation function were the identity function rather than the hyperbolic tangent, then the internal states resulting from the delay line would simply correspond to delay coordinates with a delay variable equal one and an embedding dimension equal to the network size.

After having generated the weights, the dynamic rule of the internal states given by equation 3.8 is integrated, where the Lorenz input data is used as the driving signal. We have observed that the typical time interval needed for this driven regime is approximately ten Lyapunov units. This may be interpreted the time needed to explore the chaotic attractor. Once the driven regime is performed, the output weights are calculated via equation3.9. An example of the resulting prediction are shown in Figure 3.7. To generate the two first plots, we have performed the above described procedure for two different values of the scaling range of the random weights.

The observations in the previous paragraph give rise to several questions related with the choice of the parameter setting that defines reservoir computers. We have seen that the setting is defined via three weight matrices W, W_{in} and W_{out}, where W and W_{in} are random and W_{out} is linearly optimized. The first evident idea that may come to mind is to optimize the weights W and W_{in}. In fact, the setting would in that case be very similar to standard recurrent neural networks. The main difficulty that would arise in that case is the non-linearity of the loss function in dependence of the weights. While in the linear case we have been able to find the optimal solution of weights in closed form, minimizing a nonlinear loss function may be quite difficult. Standard approaches for this include for example the back-propagation method, which is effectively an update rule for the weights based on the error made during the learning phase. This method can be difficult when the depth of the network, it means the number of layers in the network architecture, becomes large. For this reason it is rather unusual to employ back-propagation methods when using recurrent neural networks. Also, for our situation, it is rather unwanted to use back-propagation because we can empirically observe that the predictive power of reservoir computers can be achieved with linear optimization. Therefore, a more suited question would whether there are some general guiding principles of how to generate the random weights and how to define their network topologies and scaling ranges, such that predictive power is attained. The simplest network topology that one could try is the case where the internal weight matrix is a diagonal matrix. This would mean that the nodes that constitute the reservoir network are not connected. The defining equation of the evolution of each node involves only the state variable of that node together with the driving signal. The reason why

such a setting is unlikely to work can be seen by considering the case where the driving signal tends to zero. In that case, that map $y_{t+1} = \tanh(\beta y_t)$, which would be the defining equation of each node, admits two attracting fixed points at ± 1 and one unstable fixed point at the origin if the slope β of the activation function is superior than one, and admits only one stable fixed equal to zero if the slope is smaller then one(Figure 3.8). The internal dynamics tend therefore to fixed points in the two cases and makes therefore the sustainment of oscillations difficult. The sustainment of oscillations is important because the internal states must evolve autonomously in the prediction phase.

The next simple attempt to find an internal network topology that implies predictive performance is to connect the nodes each one with its neighbour such that a line is formed. This may evidently increase the dimension of the dynamical state space of the internal states and thus may allow for a more diverse set of dynamical regimes to be attained. Building a delay line network is equivalent to setting all the internal connection weights to zero except those that are located directly below the diagonal(Figure3.9). Aside to the increase of the possible dynamical regimes this network allows to ensure for property called memory. Saying that a specific network exhibits memory means is saying that in the instantaneous value of the internal state is encoded information about the values of the observational signal several steps back in history. The more steps backward in history that are encoded the larger the memory of the network is. For our purposes, which consist of modelling non-linear dynamical data, the need for memory is obvious because most real-world signals are only partial and thus require some kind of reconstruction before being modelled. This difficulty is alleviated if the network exhibits a sufficiently long memory to capture the dynamics. The advantage here is that the question of which points from the history need to be taken into account, this means, how large must the delay time be and which embedding dimension should one use, these questions, are answered by solving the linear regression problem for W_{out} without having to perform tedious visual inspections routines to find the delay time or the embedding dimension. A major distinction here is that the internal states are only representations of the observational dynamical variables, they encode the necessary information, but they are not equivalent to delay coordinates.

Having had an idea on how the internal network topology may affect predictive performance, we may consider the determination of the outer layer W_{out} in more detail. To recapitulate, the approach was to generate a time-series of internal states $\mathbf{y}_t$ which is driven by the observational data $\mathbf{x}_t$ with help of the dynamic rule defined by the activation function. Once we have generated such a series, we look for a set of weights W_{out} that associate to each value of the internal state the value of the observational data at the corresponding point in time. Thus, W_{out} can be identified as a function that allows to extract the information encoded by the representations of the data that consist of the internal states. In order to find this set of weights, we have written down the error that would have been made by the W_{out} during a time interval for which we know the answer, this means

$$\varepsilon =< (W_{out}\mathbf{y}_t - \mathbf{x}_t)^2 >_T \tag{3.12}$$

For simplicity, we take the variables $\mathbf{y}_t$ and $\mathbf{x}_t$ to be scalars and thus the readout weights W_{out} would simply be a number. In this situation, we define the optimal readout simply

as the one that minimizes the regression error ε. The value of W_{out} that minimizes the regression error ε can be found by setting the derivative of the regression error with respect to the weights equal to zero. Thus we have,

$$\partial_{W_{out}}\varepsilon =< \mathbf{y}_t(W_{out}\mathbf{y}_t - \mathbf{x}_t) >_T = 0 \tag{3.13}$$

which yields

$$W_{out} =< \mathbf{y}_t\mathbf{y}_t >_T^{-1} < \mathbf{y}_t\mathbf{x}_t >_T = 0 \tag{3.14}$$

The previous equation was derived for scalars but it does also hold for higher-dimensional variables but would involve in that case a matrix inversion procedure. Thus, the main easiness of the linear dependence of the loss function on the weights is the closed form solution. Having a linear dependence of the loss function on the readout weights does necessarily imply that the readout must be a linear function of the internal states. For example, we could have taken a function that includes higher orders in the internal states $q(\mathbf{y}_t)$ of the internal states and tried to extract the informational about the observational data from it as $\mathbf{x}_t = W_{out}q(\mathbf{y}_t)$. While it might not be immediately clear why higher order in $\mathbf{y}_t$ may be relevant, it is intuitively clear that allowing more orders to contribute can enhance performance. The reason why a nonlinear readout function may be needed can be seen by considering the form of the activation function. We have taken the activation function to be a sigmoid, which means a function that tends to ± 1 when its argument tends to $\pm\infty$ and in the transition region around the origin is approximately linear. This form implies that the range of the internal states is the interval $[-1, 1]$. If the slope of the activation is large, then the form of the activation function consists of two plateaus near the boundaries of $[-1, 1]$ and a steep increase in between. In that case, most of the values will be mapped either to 1 or -1, and it is conceivable that in the density distribution function of the internal state two large peaks will build at those points and the density will be low in between. Because most of the observational data we get from the real world have most likely Gaussian distribution, or small deviations from at the tails, and because the Gaussian admits one peak at the center and decreases for large values, it conceivable that the relation between the observation data and internal states is most likely nonlinear, otherwise it is difficult to distort the Gaussian distribution into a distribution with two peaks at the boundaries of the definition domain and small values at the center(Figure 3.10).

Having made simple consideration about what may be important for the structure of the weight matrices W_{in}, W and W_{out}, the next question that one may ask is the range of scaling of these matrices. To make the problem clear, we consider again the case of $\mathbf{y}_{t+1} = \tanh(W\mathbf{y}_{t+1})$. We have seen in the scalar case how the value of W may affect the stability of the fixed point $\mathbf{y} = 0$. It seems therefore intuitive that by changing the range of scaling of the internal weight matrix the internal weights dynamics may change. To change the range of scaling, we first generate the internal weight matrix W by elements that are either zero or one, corresponding to either connected or disconnected nodes. This first determines the topology of the network, and is typically done either by random algorithms, like the random Erdös-Reni network, or by constructing a delay line network as discussed

in the previous paragraph. One the binary connections are generated, for each connection we draw a random number from the interval $[-\beta, \beta]$ and associate it to the connection. If the matrix element associated to a particular connection is zero, it will remain zero after the random numbers procedure. For the internal weight matrix a similar procedure can be made, but without recurring to random graph methods because the matrix not a square matrix. The empirically good choice for the generation of the internal weight matrix is to inject to each node from the network one particular component from driving signal. After having generated the topology of the internal weight matrix, essentially the same procedure as above is made to find the random weights from an interval $[-\alpha, \alpha]$. Now the question is about the choice of α and β. The first thing that may come to mind is to perform a gradient descent but now not only with respect to the readout weights but also to the two scaling variables α and β. If we were to optimize with respect to these variables, then the optimization would be nonlinear since these variables appear within the sigmoid. So an alternative would be to choose a specific value for the pair of (α, β) and run the learning phase for that specific pair and inspect the the error obtained from the linear regression. By doing this for may values of (α, β) we are effectively optimizing these two variables but now the optimization problem is decomposed into a linear problem, which involves the determination of the readout weights, and the pair (α, β), which involves a non-linear loss function. Aside to the possibility of inspecting the change of the error made within the learning phase, which is the regression error, we can in a similar way do the same procedure but with inspecting the error during the prediction phase. This corresponds to feeding the output consisting of the predictions obtained from the internal network states back into the input and repeating the same procedure. This is distinct from the regression error, since now we are evaluating the error with respect to observational data that the internal network has not seen. By means of this validation error we are therefore inquiring the ability of the network to generalize, while with the regression error we are testing the ability of the network to memorize. Below we document the results obtained for the validation error as seen in the plane defined by the pair (α, β)(Figure 3.11, Figure 3.12). These results relate to the case that all the network parameters are held fixed and only the pair (α, β) is varied. It is seen that there are clear correlations between these parameters and their change clearly affects the performance. An important point here is that performance was defined as the validation error during approximately ten Lyapunov units, where one Lyapunov unit is the inverse maximal Lyapunov exponent of the observational data. The reason for this choice is that we expect the network prediction to match the true trajectory only for period of time equal to several Lyapunov exponents. If we define the time interval much larger than this, the two trajectories will certainly separate at some point and the errors made from that point will make the correct predictions undistinguishable from the incorrect ones. In turn, if we choose the time interval for the validation error much smaller than the Lyapunov unit, the validation error would be very small since the prediction usually do follow the true trajectory even for a very short time. Thus, the parameter study presented here relates to short-term predictive power of the reservoir network. We also insist that, while we inquire the variations of the short-term predictive power, in practice it is not possible to have predictions that do not have short-term fluctuations. This is because the local

Lyapunov exponent, which quantifies predictability near the point in phase space that is under consideration, admits short-term fluctuations. So by changing the initial value of the observational data one may expect that the short-term performance may change. It is however a priori not clear, how the structures indicated in the Figure below are affected by the change of initial conditions.

3.5 Reservoir computers for Spatio-temporal systems

The function approximation properties of neural networks are in general known to improve as the number of internal weights is increased. In our case, this corresponds to choosing the dimension of the internal states orders of magnitudes larger than the dimension of the observational data. Since the time complexity of matrix inversion operations is $O(n^3)$, this would render the estimation of the readout function computationally ineffective for large signals, such as those arising when treating spatio-temporal chaotic systems. So how can we apply reservoir computers to data having a large dimension ? One possible approach that has been proposed by [13] and that makes reservoir computers applicable to large multivariate data is to exploit the locality of the underlying dynamics. This means that the future of each data point at a given time is only affected by the its neighbours at that specific time. For physical applications this is implied by the fact that physical interactions are often short ranged. To take advantage of the local dynamics, the observation data is subdivided into spatially contiguous variables, where each of these is associated to a separate internal network(Figure 3.13). For an internal state variable with dimension n that is subdivided into several local internal variables with dimension m each, this reduces the time complexity from $O(n^3)$ to $O(m^3)$, $m \ll n$ Since the local internal variables are evolved in parallel, this scheme is computationally effective in view of modern computer architectures involving massively parallel processing units.

3.6 Performance of reservoir computers

In what follows, we briefly document the main lines of our implementation. We then present the results of the performance study and discuss its implications on the calibration of reservoir computers.

We consider the situation where we are given a series of observational data $\{\mathbf{x}_t\}$ for $t \in [0, T]$ of a variable $\mathbf{x}_t \in \mathbb{R}^N$ that we assume to obey a deterministic dynamical equation of the form

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t) \tag{3.15}$$

for some unknown $f : \mathbb{R}^N \to \mathbb{R}^N$. From the recorded past values for $t \in [0, T]$ we wish to infer future $t \geq T$ to high accuracy until a certain lapse of time that characterizes the dynamical system in question. After this time it is only desired that the forecast remains qualitatively similar to the correct behaviour of the system. We will namely address cases where $\mathbf{x}_t$ evolves on a chaotic attractor, implying an exponential temporal growth of errors in the initial conditions, making the inference task challenging. While the direct way of inference would be to approximate f by a linear combination of a suitably chosen functional basis, we proceed by introducing the dynamical variable $\mathbf{y}_t \in \mathbb{R}^D$ that is coupled to the observational data through $g : \mathbb{R}^D \times \mathbb{R}^N \to \mathbb{R}^D$ according to the driven dynamical system given by

$$\mathbf{y}_{t+1} = g(\mathbf{x}_t, \mathbf{y}_t) \tag{3.16}$$

In particular, for a suitably chosen coupling g the dynamical system defined by the set of equations (1) and (2) may achieve an asymptotic state of invertible synchronization[16, 17], where the driving and the driven variables $\mathbf{x}_t$ and $\mathbf{y}_t$ are related by a static functional relation $\varphi : \mathbb{R}^D \to \mathbb{R}^N$ such that $\| \varphi(\mathbf{y}_t) - \mathbf{x}_t \| \to 0$ for $t \to \infty$. Once synchronization is established the inference is reduced to finding a suitable approximation for φ, allowing to evolve the internal state dynamics y_t according to the autonomous equation,

$$\mathbf{y}_{t+1} = g(\varphi(\mathbf{y}_t), \mathbf{y}_t) \tag{3.17}$$

and to infer the variable of interest from it systematically through $\mathbf{x}_t = \varphi(\mathbf{y}_t)$. Whether synchronization is established can be deduced by monitoring the correlation matrix $< \mathbf{y}_{t'}^\top \mathbf{x}_{t'} >_t$ as time t increases in $[0, T]$(see Figure 3.15a). A correlation that settles on a constant value and does not change as time changes is a hint that the relation between $\mathbf{x}_t$ and $\mathbf{y}_t$ is independent of time. Once the correlation has settled the variable change φ can be extracted from it through the leading order estimation $\varphi(\mathbf{y}) =< \mathbf{y}_t^\top \mathbf{y}_t - r\mathbb{I} >_T^{-1} < \mathbf{y}_t^\top \mathbf{x}_t >_T \mathbf{y}$, which results from the regularized regression of $\varepsilon =< \| \varphi(\mathbf{y}_t) - \mathbf{x}_t \| >_T$ with respect to φ. In principal, a uniformly contracting coupling $\| g(\mathbf{x}, \mathbf{y}_1) - g(\mathbf{x}, \mathbf{y}_2) \| \leq \| \mathbf{y}_1 - \mathbf{y}_2 \|$ ensures an initial difference between two internal states driven by the same series $\{\mathbf{x}_t\}$ to be made arbitrarily small for a sufficiently long series[16]. This ensures the internal states $\mathbf{y}_t$ to be uniquely related to the observational series $\{\mathbf{x}_t\}$. For an internal state dimension at least twice as large as the dimension of observational data, this relation is likely to be invertible[16], thus implying the establishment of the indicated state of synchronization.

Uniform contraction can be for example imposed for a sigmoidal coupling

$$g(\mathbf{x}, \mathbf{y}) = \tanh(W_{in} \cdot \mathbf{x} + W \cdot \mathbf{y}) \tag{3.18}$$

by choosing the largest singular value of W smaller than one, where $W_{in} \in \mathbb{R}^{D \times N}$ and $W \in \mathbb{R}^{D \times D}$ are two random matrices whose entries are uniformly sampled from the intervals $[-\alpha, \alpha]$ and $[-\beta, \beta]$, respectively. In practice, this condition was shown to be restrictive in the sense that it excludes a good fraction of W possibilities that imply well defined internal states[18]. A formally less plausible yet practically more favoured condition consists in tuning the entries range β of W close to the unity[19].

It has been recently demonstrated that by changing the entries range α of W_{in} the stability properties of the invariant set $M = \{(\mathbf{x}, \mathbf{y}) : \varphi(\mathbf{y}) = \mathbf{x}\}$ may change[16]. Stability here relates to transversal perturbations $\varepsilon_t = \| \varphi(\mathbf{y}_t) - \mathbf{x}_t \|$ away from the invariant set M during the autonomous dynamical regime, rather than the stability of internal states $\mathbf{y}_t$ when driven by a certain $\{\mathbf{x}_t\}$ as discussed above. The two regimes are fundamentally distinguished by the approximation errors in the drive signal made by the leading order approximation of φ. For a stable M these errors will drive the forecast away from the true signal after a short term while the forecast will maintain statistically consistent properties in the long term. In turn, an unstable M implies also a correct short-term forecast but the long-term behaviour would be inconsistent in this case. Thus, the change of stability properties of the invariant set M clarifies the variability of the long-term forecasting quality. It stays a priori unclear why the short-term forecasting quality shows strong sensitive dependence on the parameters[20]. In particular, this sensitivity was observed for parameter regions where the long-term forecasts remain qualitatively unchanged, i.e where the set M is stable. This fact indicates the presence of a parameter window separating stability from instability.

Chaotic dynamical systems of which the attractor A is constrained to a subset M of the full degrees of freedom are known to exhibit bifurcations where the attractor first looses transversal stability only in the neighbourhood of its unstable periodic orbits of low period[21], and then becomes fully transversally unstable[22, 23]. Within such a regime, a typical chaotic trajectory subject to transversal perturbations away from M would explore the attractor A for long times until it falls into the neighbourhood of a transversally unstable orbit where the transversal errors get magnified. For a globally attracting M this mechanism results in the appearance of intermittent bursts[24] where chaotic trajectories of A are on the one hand repelled from M by the transversally unstable orbits and on the other hand attracted back to M due to the global dynamics of the system. We propose this as a basis to understand the intermittent bursting behaviour appearing in the transversal error $\varepsilon_t = \| \varphi(\mathbf{y}_t) - \mathbf{x}_t \|$ during the driven dynamical regime. Evidence for our proposal is the scaling law[22] describing the mean interburst time τ as well as the maximal bursts amplitude ε_{max} when the parameter α is varied.

The bursting behaviour is illustrated in Figure 3.15b for the three chaotic flows f used in our study[25]. The plots show the time evolution of the transversal error ε_t during the driven regime. As the parameter α is varied, the bursts maximal amplitude decreases as

System / Scaling law	Lorenz	Halvorsen	Rößler
$\log f(\epsilon) \sim \log \epsilon$	0.038	0.039	0.038
$\alpha \log \tau \sim \log \alpha$	0.89	0.98	1.34
$\varepsilon_{max} \sim \alpha$	$8.1 \cdot 10^{-4}$	$1.1 \cdot 10^{-3}$	$3.7 \cdot 10^{-4}$
α_c	0.26	0.15	0.15
λ_{max}	0.9056	0.7899	0.0714

Table 3.1: Summary of the computed scaling laws and the corresponding scaling exponents.

a linear function of α, until it takes a low constant value corresponding to the standard deviation of the error (Figure 3.15c), where high quality synchronization is achieved. This qualitative change in the bursting behaviour can be monitored by determining the mean duration between two consecutive bursts τ, which is displayed in Figure 3.15d. The scaling shown is obtained by fixing the random samples of W and W_{in} and evaluating ε_{max} and τ at equidistant values of α starting from the critical point α_c, which we define as the point where pronounced scaling begins starting from $\alpha \ll 1$ and thus identify as the cutoff value where linear scaling is observed in Figure 3.15c. The critical value α_c for each data set can be found in Table 3.1. We note that different scaling exponents for the interburst time are expected from the fact that the three data sets have different Lyapunov spectra, thus corresponding to different repulsion rates in the direction parallel to M[22].

The evidenced locally repellent sets within the basin of attraction of the invariant set M have direct consequences regarding tuning the reservoir. By tuning is meant the problem of specifying a pair of values (α, β) for given random matrices W_{in} and W such that the short term forecasting performance is optimal. Optimality is achieved when the duration of accurate short-term forecasts is evidently maximized, and importantly when this duration is not significantly affected by the finite time segment $\{\mathbf{x}_t\}$ learned by the reservoir. Indeed, since the bursts of ε_t during the driven regime do not alternate the regression error ε due to their short lived and infrequent nature, the inspection of ε as a function of (α, β) leads to a structureless landscape. In contrast, starting the autonomous regime at a time $t\prime$ in the order of ten Lyapunov units earlier than T and inspecting the average $\varepsilon\prime$ of ε_t within this autonomous regime in $[T - t\prime, T]$ yields a highly structured pattern. Figure 3.17a shows an example for the Lorenz chaotic system. The shown patterns are highly dynamic in the sense that a slight shift of the initial point $\mathbf{x}_0$ of the learned time-series results in clear change of the pattern, as illustrated by Figure 3.17b.

Thus, the difficulty we encounter when trying to extract the optimal pair (α, β) from the resulting $\varepsilon\prime$ landscape is that this landscape relates only to the narrow time interval where it was obtained from, i.e the resultant optimal parameter sets indicate transversal stability only in a narrow portion of M. The variations of the transversal stability of the synchronization manifold as well as its existence can be evidenced by calculating the transversal Lyapunov exponents. Since the defining equation of the autonomous regime for the internal states is known, we proceed by calculating the Lyapunov spectrum of

the internal states during the autonomous dynamical regime according to the variational equation,

$$\delta_{t+1} = J_g(\varphi(\mathbf{y}_t), \mathbf{y}_t)\delta_t \tag{3.19}$$

where δ_t refers to a small perturbation of the internal state. The transversal Lyapunov exponent λ^t_{max} is identified as the maximal Lyapunov exponent from the above calculated spectrum after elimination of the Lyapunov exponents of the driving signal(we refer to [16] for a detailed justification). The variations of λ^t_{max} as a function of α and β are shown in Figure 3.17(c-d). While Figures 3.17(a-b) and Figures 3.17(c-d) show some kind of similarity, a substantial difference is the presence of fine scaled alternations in the former case. Notably, the fine scaled alternations were observed in the performance landscape related with several driving signals. While the mechanism underlying these fine scaled alternations is intricate, their presence immediately indicates the difficulty of tuning (α, β) with respect to $\varepsilon\prime$.

We insist that this difficulty becomes critical if these fine scaled alternations have a fractal nature[26]. We mention that fractal parameter sets are already found in the simplest possible nonlinear equation, namely the quadratic map, where arbitrarily close to a given parameter value for which chaotic solutions are typical, one can find a parameter value that yields periodic solutions. In a similar way, we classify our parameter pairs (α, β) into two classes of optimal or suboptimal choices depending on whether $\varepsilon\prime$ is below or above a certain quality threshold δ. We suggest that arbitrarily close to a any given pair (α, β) belonging to the one class one can find a pair that belongs to the other. Evidence of such a pathological topology can be obtained by calculating the uncertainty exponent γ of the defined parameter sets. The uncertainty exponent characterizes the scaling behaviour of the fraction $f(\epsilon)$ of uncertain parameter points as the parameter precision ϵ is varied. A parameter point α is called uncertain with respect to given precision ϵ if the points obtained by shifting that point $\alpha \pm \epsilon$ by the given precision belong to a different class. Figure 3.19 illustrates the alternations with magnified pictures, as well as the scaling behaviour of the uncertainty function(Figure 3.19d). The indicated scaling shows examples obtained from performing the described procedure for different initial data points $\mathbf{x}_0$ (Figure 3.19e), showing that the scaling law $f(\epsilon) \sim \epsilon^\gamma$ is indeed robust. Notably, for internal states driven by several distinct chaotic systems the resultant value of the uncertainty exponent $\gamma = 0.038 \pm 0.002$ occurs to be the same within the indicated error tolerance. The resulting exponents are summarized in Table 3.1. An implication of a scaling exponent $\gamma < 1$ is that increasing the precision does not significantly decrease the uncertainty.

Thus, we suggest the well known extreme sensitivity of reservoir computers to originate from a seemingly very generic structure of the parameter space. One consequence of such highly sensitive parameters is the inefficiency of gradient methods for finding the optimal pair (α, β) which minimizes the short-term error $\varepsilon\prime$. One way to reduce this sensitivity is to tune the reservoir by minimizing the error $\varepsilon\prime$ averaged with respect several initial points $\mathbf{x}_0$. An other way is to maximize the interburst mean period τ, which is by definition a long-term average and thus stable against finite time fluctuations. To show the equivalence between the two methods, we determine the optimal parameter pair (α, β) resulting from

the error $\varepsilon\prime$ averaged over a number of 30 initial data points $\mathbf{x}_0$, as well as the parameter pair for which the interburst time τ is maximal. We then compare the two optima by evaluating for each optimum the mean and standard deviation of the resulting forecast horizon with respect to a number of 10^4 consecutive initial data points $\mathbf{x}_0$. The comparison is shown in figure 3.20, underlining the equivalence of the two described calibration methods.

To conclude, we have characterized the learning process as the establishment of a regime of strong temporal correlation between the observational and internal dynamical states. We have characterized the high-amplitude bursts interrupting this regime and investigated the corresponding consequences related with tuning the parameters (α, β). We have noted that, while (α, β) do not seem to affect the performance if performance is evaluated by means of the regression error ε, the inspection of τ and ε_{max} as a function of (α, β) shows clear variations. The quantities τ and ε_{max} may therefore serve as metrics for optimizing (α, β). Further, while these considerations were related to the driven dynamical regime, we have noted the difficulties that may arise when defining performance via the error $\varepsilon\prime$ obtained from the autonomous regime. We have quantified the sensitivity of $\varepsilon\prime$ by calculating the corresponding uncertainty exponent. The precise mechanism underlying the fine scaled structures, as well as the reason for the seemingly generic value of the uncertainty exponent, have remained unclear. Future work shall be dedicated to these questions.

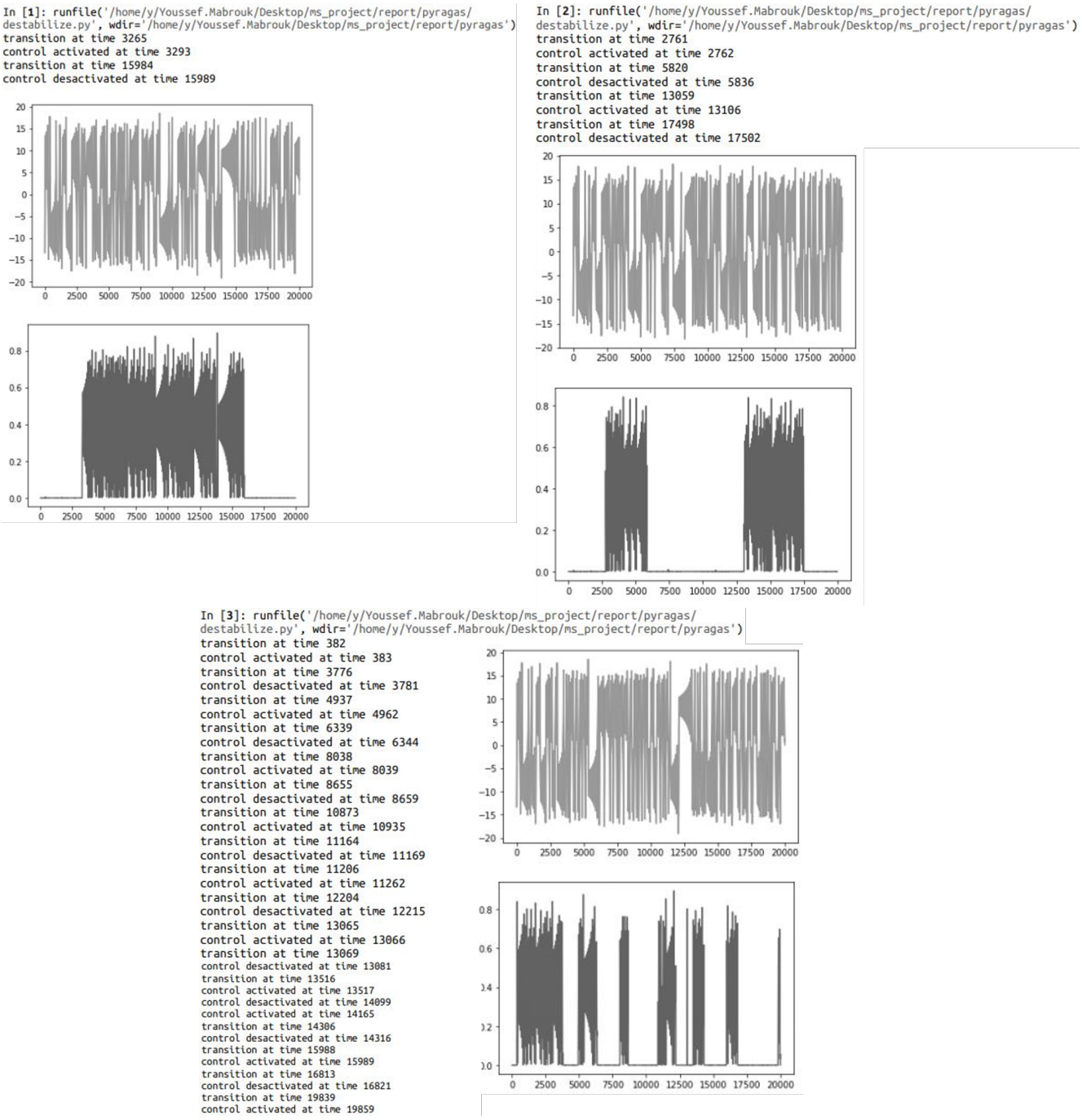

Figure 3.6: Detecting and controlling transitions to transient chaotic state. The panels indicate three runs of the above described procedure for different transition probabilities. The program prints the time at which the random transition takes place as well as the times where the control is switched off and on. In each panel, the first plot shows the time resolution of one component, and the second plot shows the time evolution of the control error which serves as indicator to switch the control off and on. The printed time points of the transitions and switching times indicate that the peaks in the control error function well as indicators.

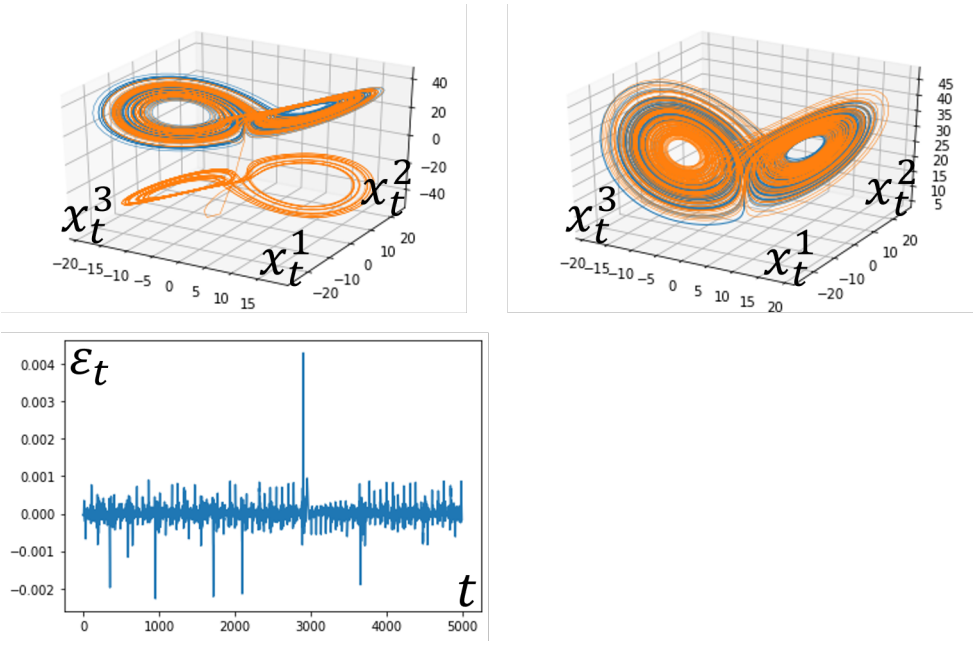

Figure 3.7: Performance of reservoir computers. The first two panels show a superposition of the trajectory obtained from the learned model and the correct trajectory for two parameter value. The third panel illustrates the time resolution of the regression error.

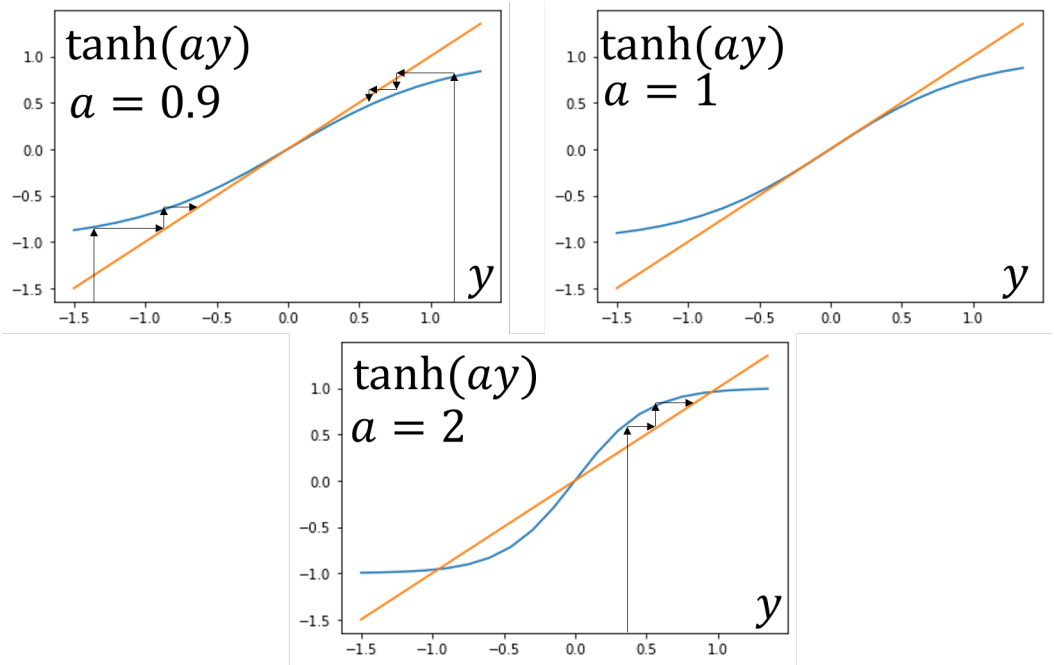

Figure 3.8: Bifurcation of the fixed point at the origin for the dynamics given by the activation function for one single node.

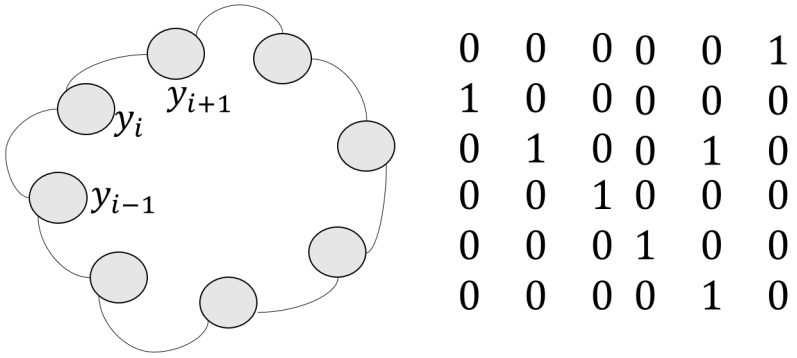

Figure 3.9: Delay line network illustrated in matrix representation for a network with six nodes. The shown number indicate the matrix elements of the internal weight network W. The graph representation illustrates the corresponding connections between the nodes.

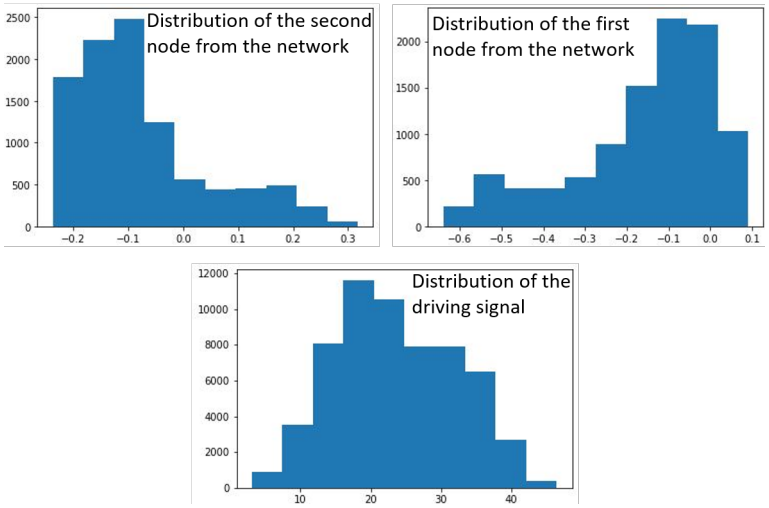

Figure 3.10: Comparison between the densities of the driving signal and the density of the network nodes. The first two panels show the typical densities that are obtained by taking a histogram of one node from the network. The third panel shows the distribution of the third component of the driving signal, which is in this the Lorenz signal, which represents the typical case of a Gaussian distribution.

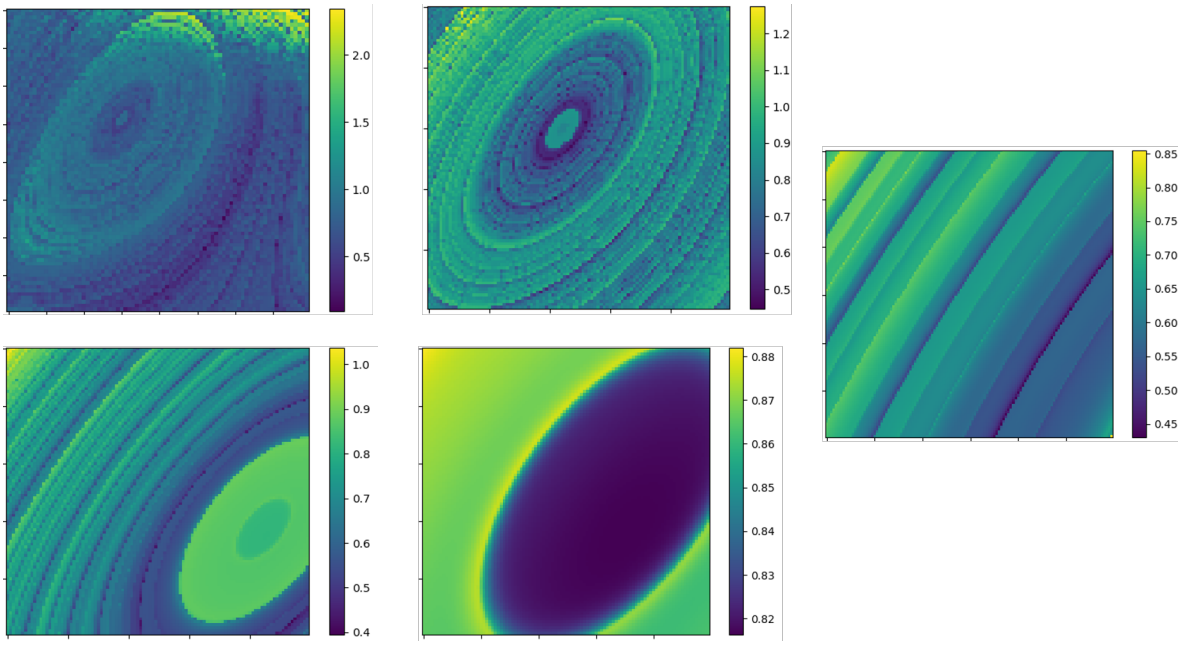

Figure 3.11: Performance landscape represented by the contour plot of the validation error in the (α, β) plane. In the first plot from the upper left the scaling range of the variables α and β correspond to the interval $[-1, 1]$. From left to right the plots are related to smaller and smaller grid search regions, where each time the indicated region is observed with higher resolution. The same delay line architecture was used for all the plots. For each grid point an integration of the equations of motion of the internal states for $T = 10^4$ time steps was used. The same delay line architecture was used for all the plots.

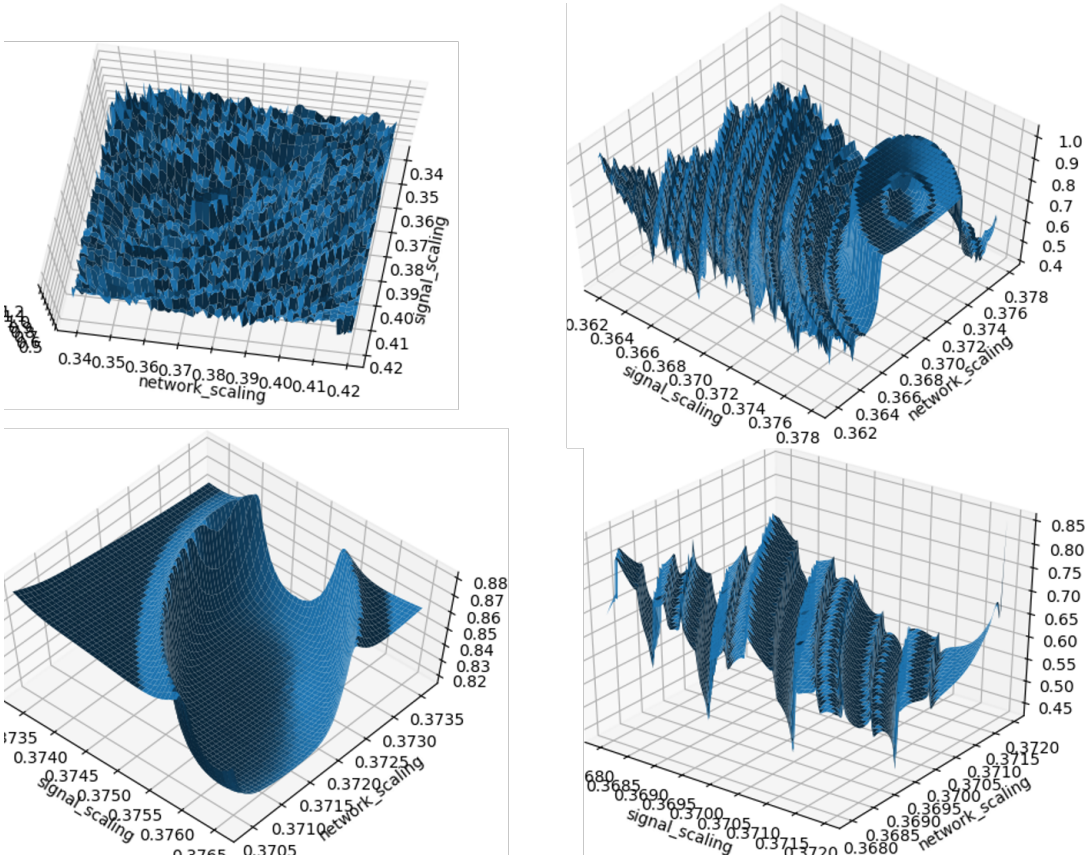

Figure 3.12: Performance landscape represented by the three dimensional plot of the validation error in the (α, β) plane. In the first plot from the upper left the scaling range of the variables α and β correspond to the interval $[-1, 1]$. From left to right the plots are related to smaller and smaller grid search regions, where each time the indicated region is observed with higher resolution. The same number of samples consisting of 40×40 grid points was used. For each grid point an integration of the equations of motion of the internal states for $T = 10^4$ time steps was used. The same delay line architecture was used for all the plots.

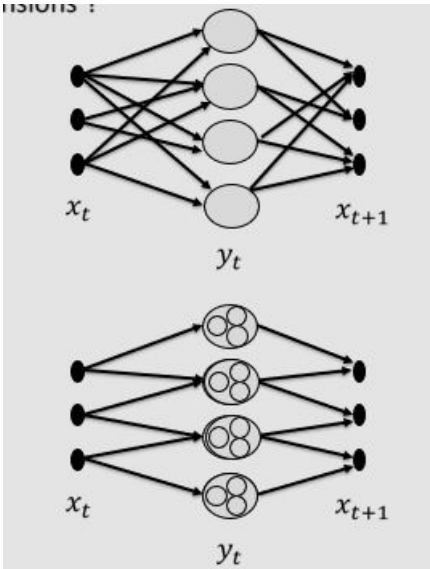

Figure 3.13: Illustration of the basic idea behind the local state approach for the treatment of spatio-temporal data.

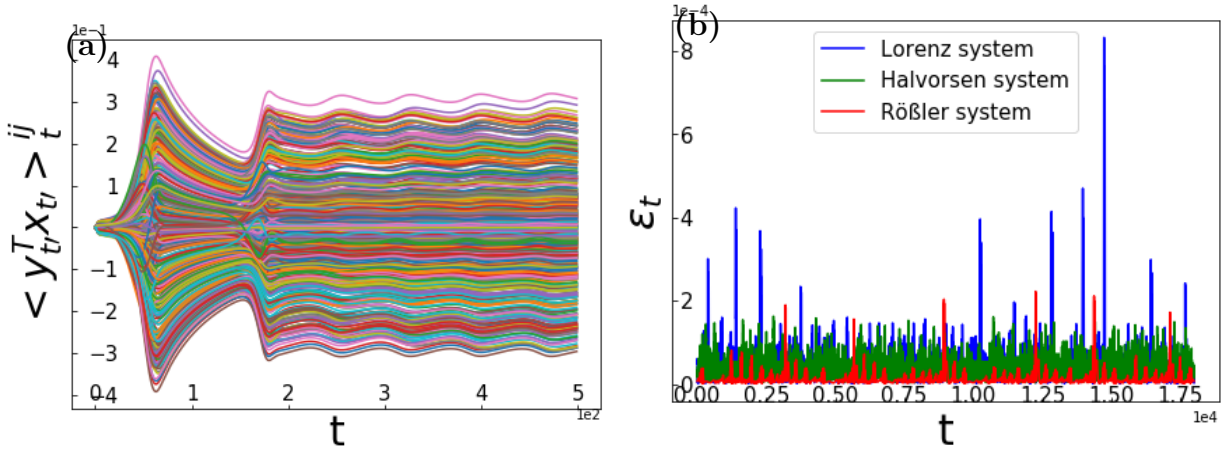

Figure 3.14: Dynamics in the driven regime. Each data set $\{\mathbf{x}_t\}$ is normalized to zero mean and variance one. For all data sets, the internal states dimension is set to $D = 500$, the internal connectivity matrix W is obtained by the random Erdös-Reny algorithm with edge creation probability $p = 2 \times 10^{-1}$, the regularization parameter is $r = 10^{-6}$. (a) Initial transient time evolution of the correlation matrix elements exemplified by the Lorenz system. (b) Long-term time evolution of the transversal error, characterized by short lived high amplitude bursts. The time interval T corresponds to 100 Lyapunov units, where one Lyapunov unit corresponds to the inverse maximal Lyapunov exponent $1/\lambda_{max}$.

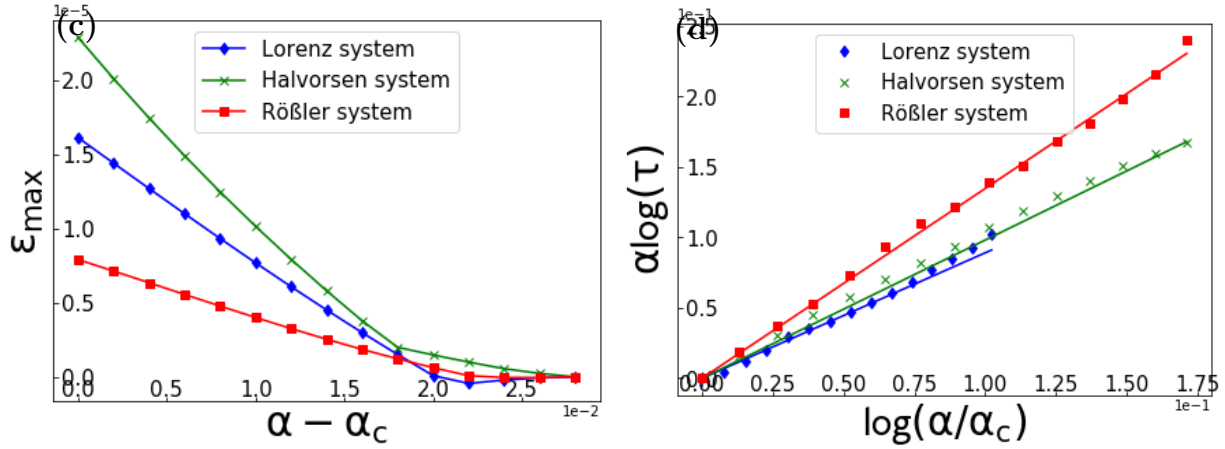

Figure 3.15: Dynamics in the driven regime. Each data set $\{\mathbf{x}_t\}$ is normalized to zero mean and variance one. For all data sets, the internal states dimension is set to $D = 500$, the internal connectivity matrix W is obtained by the random Erdös-Reny algorithm with edge creation probability $p = 2 \times 10^{-1}$, the regularization parameter is $r = 10^{-6}$. (c) Maximal bursts amplitude ε_{max} as function of α. (d) Interburst duration τ as a function of α. The sampling resolution of α is 2×10^{-3}.

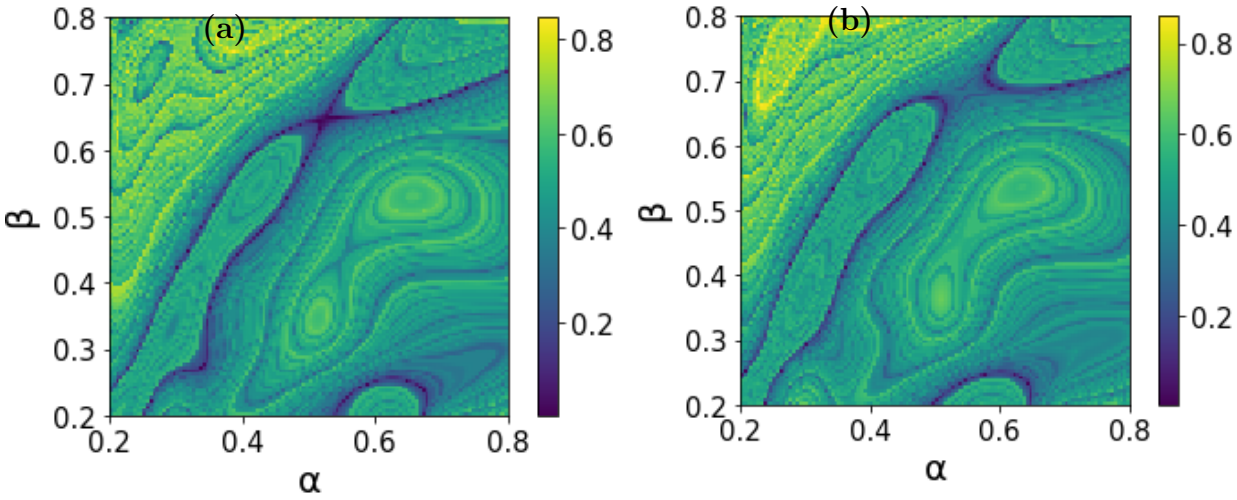

Figure 3.16: Performance landscape $\varepsilon\prime$ within $t\prime = 10$ Lyapunov units of autonomous evolution after a duration $T - t\prime$ corresponding to 100 Lyapunov units of driven evolution. A parameter grid of $10^2 \times 10^2$ was used to generate each of the shown results. (a) $\varepsilon\prime$ (b) Delayed version of (a). The dynamic change can be visualized in the Supplemental Material via Videos showing the patterns for several consecutive values of $\mathbf{x}_0[2]$.

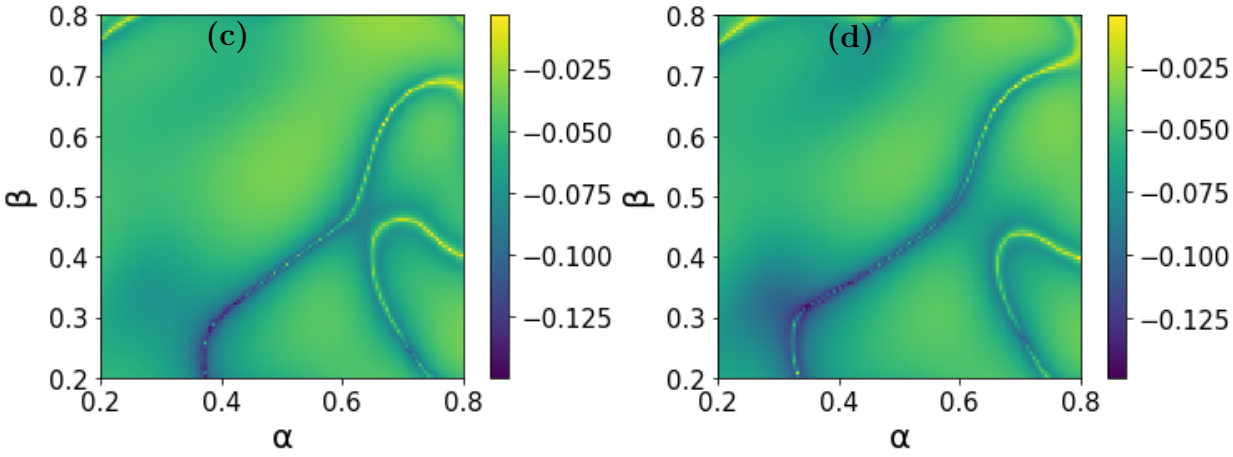

Figure 3.17: Transversal Lyapunov exponent λ^t_{max} within $t\prime = 10$ Lyapunov units of autonomous evolution after a duration $T - t\prime$ corresponding to 100 Lyapunov units of driven evolution. A parameter grid of $10^2 \times 10^2$ was used to generate each of the shown results. (c) λ^t_{max} (d) Delayed version of (c). The dynamic change can be visualized in the Supplemental Material via Videos showing the patterns for several consecutive values of $\mathbf{x}_0[2]$.

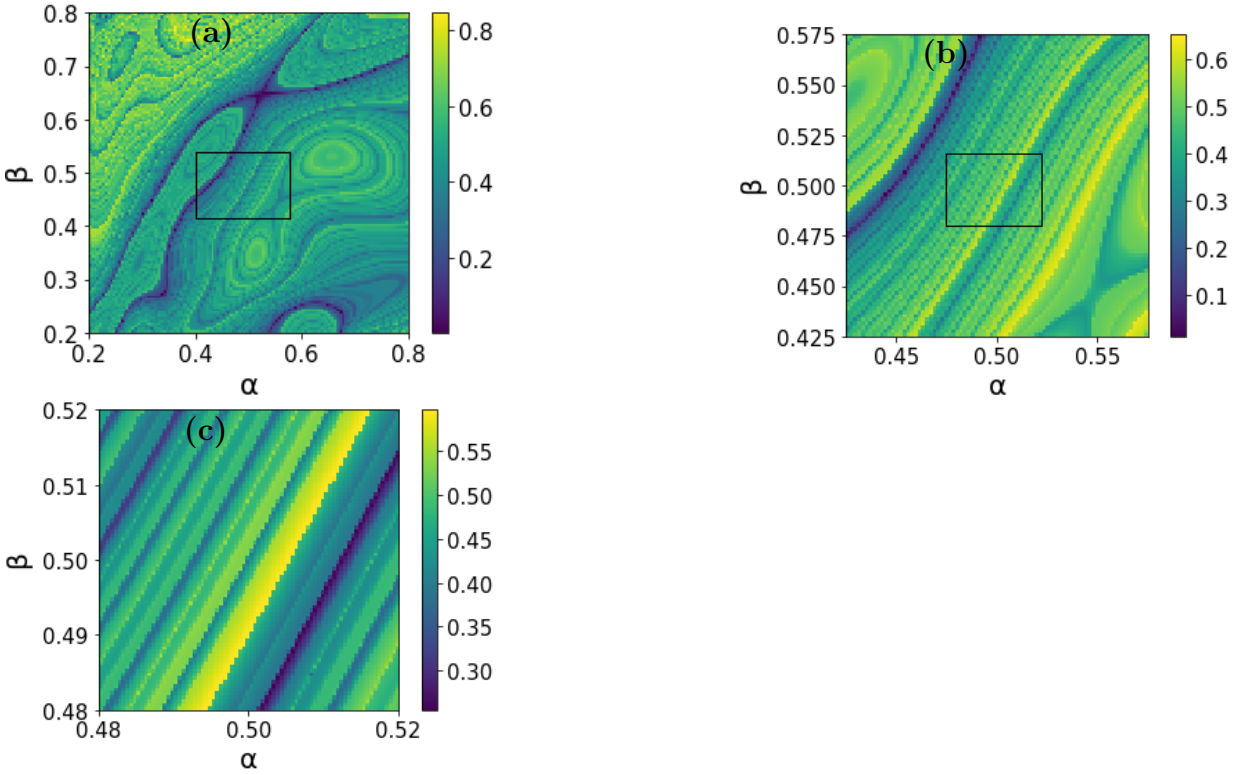

Figure 3.18: Uncertainty analysis. (a) Performance landscape of the Lorenz system. (b) and (c) Blown up versions of (a).

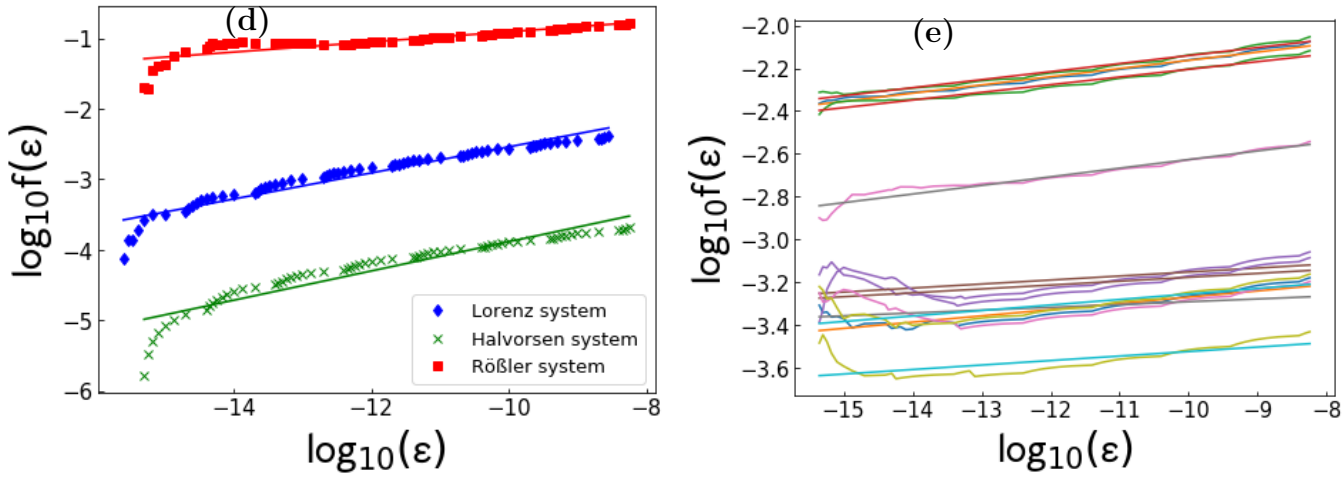

Figure 3.19: Uncertainty analysis. (d) Scaling of the uncertainty function. The horizontal axis corresponds to the logarithmic measurement precision ϵ of the parameter α and is not to be confused with the regression error ε. A parameter grid of size $10^2 \times 10^2$ in logarithmic scale was used in order to calculate each of the shown scaling behaviours. (e) Variations with respect to initials conditions exemplified for the Lorenz system.

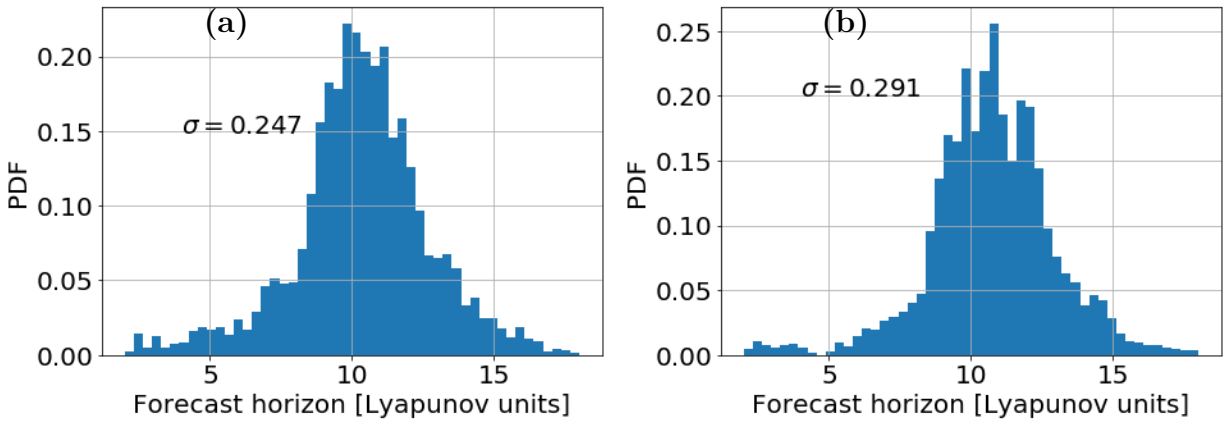

Figure 3.20: Fluctuations of the forecast horizon due to finite time fluctuations for the Lorenz system. (a) Optimum of $\varepsilon\prime$ averaged in respect to 30 initial points $\mathbf{x}_0$. (b) Optimum of τ.

Chapter 4

Conclusion

The main result of this work is to quantify the parameter uncertainty of reservoir computers. The method we have found convenient is the calculation of the uncertainty exponent. The calculation resulted in an exponent close to zero. The value of this exponent was found to strongly fluctuate if the range of scaling of the parameter precision does not cover nearly more than five order of magnitudes. The statement we have made about the topology of the considered parameter sets follows directly from the consideration of the parameter set topology of elementary nonlinear equations that admit a similar uncertainty exponent. The second main result of the work is to describe the dynamic interplay between the observational and internal dynamical states. While the dynamics of the full system has a dimension initially equal to the sum of the internal dimension and data dimension, we have supported the hypothesis that the full system states are attracted to an invariant synchronization manifold consisting of the observational states together with their images by the synchronization map. It has been previously shown that the synchronization manifold is invariant under the driven as well as the autonomous dynamical regimes if the synchronization map is known exactly. We have suggested that the intermittent spikes in the transversal error result from the fact that the synchronization manifold undergoes a parametric regime where it is attracting only up to a set of measure zero consisting of the invariant sets of the observational data together with their images by the synchronization map. This is strictly the only statement we have made about the structure of the basin of attraction of the synchronization manifold. This statement is supported by the scaling of the typical time period separating the spikes of the transversal error. We precise that we have not associated the fractal parameter structures to degraded predictive power. In fact, we have conveyed that the topology of optimal parameters does strongly depend on the metric used to define them. For a metric that reflects global transversal stability of the synchronization manifold, such as the mean interburst period, the parameter set topology resulting from the metric dependence on the parameters is an indicator for predictive power. But for a metric that is only local, such as the root mean squared error averaged over several Lyapunov units, the metric dependence on the parameters yields a fractal topology that does not reflect predictive power. Thus, the essence of our work is to indicate the importance of the choice of performance metric. Furthermore, the main point

was to introduce the performance metric τ.

Appendix A

Appendix

In the following, we present the ordinary differential equations used in this work to generate dynamical data. All the systems of equations presented here are solved with the fourth order Runge-Kutta method.

A.1 Lorenz system

The Lorenz system consists of an ordinary differential equation of a three dimensional variable whose dynamics is given by

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z \end{aligned} \tag{A.1}$$

The simulations were run for parameter values consisting of $\sigma = 10, \rho = 10, \beta = \frac{8}{3}$. For this values the system admits a positive Lyapunov exponent consisting of $\lambda = 0.9$

A.2 Rößler system

The Rößler system consists of an ordinary differential equation of a three dimensional variable whose dynamics is given by

$$\begin{aligned} \dot{x} &= -y - z \\ \dot{y} &= x + ay \\ \dot{z} &= b + z(x - c) \end{aligned} \tag{A.2}$$

The simulations were run for parameter values consisting of $a = 0.2, b = 0.2, =5.7$. For this values the system admits a positive Lyapunov exponent consisting of $\lambda = 0.071$

A.3 Halvorsen system

The Halvorsen system consists of an ordinary differential equation of a three dimensional variable whose dynamics is given by

$$\begin{aligned} \dot{x} &= -ax - 4y - 4z - y^2 \\ \dot{y} &= -ay - 4x - 4z - z^2 \\ \dot{z} &= -az - 4y - 4x - x^2 \end{aligned} \tag{A.3}$$

The simulations were run for parameter values consisting of $a = 1.27$. For this values the system admits a positive Lyapunov exponent consisting of $\lambda = 0.071$

Bibliography

[1] Justin S. Hardi, Tobias Traudt, Cristiano Bombardieri, Michael Börner, Scott K. Beinke, Wolfgang Armbruster, P. Nicolas Blanco, Federica Tonti, Dmitry Suslov, Bassam Dally, and Michael Oschwald. Combustion dynamics in cryogenic rocket engines: Research programme at dlr lampoldshausen. *Acta Astronautica*, 147:251 – 258, 2018.

[2] See Supplemental Material at [URL will be inserted by publisher].

[3] Pradip Laha, S. Lakshmibala, and V. Balakrishnan. Time-series and network analysis in quantum dynamics: Comparison with classical dynamics, 2020.

[4] David B. Brückner, Pierre Ronceray, and Chase P. Broedersz. Inferring the dynamics of underdamped stochastic systems. *Phys. Rev. Lett.*, 125:058103, Jul 2020.

[5] Lee-Wen Teng, Mei-Chu Chang, Yu-Ping Tseng, and Lin I. Wave-particle dynamics of wave breaking in the self-excited dust acoustic wave. *Phys. Rev. Lett.*, 103:245005, Dec 2009.

[6] Andrey Ganopolski and Stefan Rahmstorf. Abrupt glacial climate changes due to stochastic resonance. *Phys. Rev. Lett.*, 88:038501, Jan 2002.

[7] J. Robert Buchler, Thierry Serre, Zoltán Kolláth, and Janet Mattei. A chaotic pulsating star: The case of r scuti. *Phys. Rev. Lett.*, 74:842–845, Feb 1995.

[8] Edward Ott, Celso Grebogi, and James A. Yorke. Controlling chaos. *Phys. Rev. Lett.*, 64:1196–1199, Mar 1990.

[9] Wouter-Jan Rappel, Flavio Fenton, and Alain Karma. Spatiotemporal control of wave instabilities in cardiac tissue. *Phys. Rev. Lett.*, 83:456–459, Jul 1999.

[10] Ying-Cheng Lai and Celso Grebogi. Converting transient chaos into sustained chaos by feedback control. *Phys. Rev. E*, 49:1094–1098, Feb 1994.

[11] J. Doyne Farmer and John J. Sidorowich. Predicting chaotic time series. *Phys. Rev. Lett.*, 59:845–848, Aug 1987.

[12] Heinz Georg Schuster and Wolfram Just. Piecewise linear maps and deterministic chaos, deterministic chaos, 2005.

[13] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.*, 120:024102, Jan 2018.

[14] Olivia L. White, Daniel D. Lee, and Haim Sompolinsky. Short-term memory in orthogonal neural networks. *Phys. Rev. Lett.*, 92:148102, Apr 2004.

[15] U. Parlitz. Estimating model parameters from time series by autosynchronization. *Phys. Rev. Lett.*, 76:1232–1235, Feb 1996.

[16] Zhixin Lu, Brian R. Hunt, and Edward Ott. Attractor reconstruction by machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6):061104, 2018.

[17] Zhixin Lu and Danielle S. Bassett. Invertible generalized synchronization: A putative mechanism for implicit learning in biological and artificial neural systems. *arXiv1807.05214*, 2019.

[18] Ken Caluwaerts, Francis wyffels, Sander Dieleman, and Benjamin Schrauwen. The spectral radius remains a valid indicator of the echo state property for large reservoirs. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, page 6, 2013.

[19] Izzet B. Yildiz, Herbert Jaeger, and Stefan J. Kiebel. Re-visiting the echo state property. *Neural networks : the official journal of the International Neural Network Society*, 35:1–9, 2012.

[20] Alexander Haluszczynski and Christoph Räth. Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(10):103143, 2019.

[21] Brian R. Hunt and Edward Ott. Optimal periodic orbits of chaotic systems. *Phys. Rev. Lett.*, 76:2254–2257, Mar 1996.

[22] Shankar C. Venkataramani, Brian R. Hunt, Edward Ott, Daniel J. Gauthier, and Joshua C. Bienfang. Transitions to bubbling of chaotic systems. *Phys. Rev. Lett.*, 77:5361–5364, Dec 1996.

[23] Ying-Cheng Lai, Celso Grebogi, James A. Yorke, and S. C. Venkataramani. Riddling bifurcation in chaotic dynamical systems. *Phys. Rev. Lett.*, 77:55–58, Jul 1996.

[24] Daniel J. Gauthier and Joshua C. Bienfang. Intermittent loss of synchronization in coupled chaotic oscillators: Toward a new criterion for high-quality synchronization. *Phys. Rev. Lett.*, 77:1751–1754, Aug 1996.

[25] Julien Clinton Sprott. *Chaos and Time-Series Analysis*. Oxford University Press, Inc., USA, 2003.

[26] Ying-Cheng Lai and Raimond L. Winslow. Riddled parameter space in spatiotemporal chaotic dynamical systems. *Phys. Rev. Lett.*, 72:1640–1643, Mar 1994.

Acknowledgement

I thank Thomas Hubertus and Christoph Räth for welcoming me to their research group and giving me the opportunity to do my master thesis within the group. I thank Christoph Räth for his concise supervision of this thesis and for having introduced me to the field of data analysis and its significance in physics. I am grateful to my colleagues Jonas Aumeier, Sebastian Baur and Joschka Herteux for their support and companionship through constant and friendly teamwork. I am grateful to my family for taking care of me.

Declaration of Authenticity

I hereby declare that this thesis is my own work, and that I have not used any sources and aids other than those stated in the thesis.

Munich, 29/08/2020