

# Using Machine Learning to Replicate Chaotic Attractors and Calculate Lyapunov Exponents from Data.

Jaideep Pathak,<sup>1,2</sup> Zhixin Lu,<sup>1,3</sup> Brian R. Hunt,<sup>3,4</sup> Michelle Girvan,<sup>1,2,3,5</sup> and Edward Ott<sup>1,2,6</sup>

<sup>1)</sup>*Institute for Research in Electronics and Applied Physics, University of Maryland, College Park, Maryland 20742, USA*

<sup>2)</sup>*Department of Physics, University of Maryland, College Park, Maryland 20742, USA.*

<sup>3)</sup>*Institute for Physical Science and Technology, University of Maryland, College Park, Maryland 20742, USA.*

<sup>4)</sup>*Department of Mathematics, University of Maryland, College Park, Maryland 20742, USA.*

<sup>5)</sup>*London Mathematical Laboratory, 14 Buckingham Street, London WC2N 6DF, United Kingdom.*

<sup>6)</sup>*Department of Electrical and Computer Engineering, University of Maryland, Maryland 20742, USA.*

(Dated: 23 October 2017)

We use recent advances in the machine learning area known as ‘reservoir computing’ to formulate a method for model-free estimation from data of the Lyapunov exponents of a chaotic process. The technique uses a limited time series of measurements as input to a high-dimensional dynamical system called a ‘reservoir’. After the reservoir’s response to the data is recorded, linear regression is used to learn a large set of parameters, called the ‘output weights’. The learned output weights are then used to form a modified autonomous reservoir designed to be capable of producing *arbitrarily long* time series whose ergodic properties approximate those of the input signal. When successful, we say that the autonomous reservoir reproduces the attractor’s ‘climate’. Since the reservoir equations and output weights are known, we can compute derivatives needed to determine the Lyapunov exponents of the autonomous reservoir, which we then use as estimates of the Lyapunov exponents for the original input generating system. We illustrate the effectiveness of our technique with two examples, the Lorenz system, and the Kuramoto-Sivashinsky (KS) equation. In particular, we use the Lorenz system to show that achieving climate reproduction may require tuning of the reservoir parameters. For the case of the KS equation, we note that as the system’s spatial size is increased, the number of Lyapunov exponents increases, thus yielding a challenging test of our method, which we find the method successfully passes.

There have been notable recent advances in machine learning that have proven useful for tasks ranging from speech recognition<sup>1,2</sup> to playing of the game Go at a level surpassing the best humans<sup>3</sup>. In this paper, we build a machine learning model of a chaotic dynamical system using the neural computing framework known as reservoir computing<sup>4</sup>. We show that such a model can be used to deduce the most important quantifiers of the system’s chaotic behavior, namely, its Lyapunov exponents, using only limited time series measurements of the system. We envision that such artificial intelligence based models could be used to accurately capture the complex dynamics of many geophysical, ecological, biological or economic systems that are often difficult to model from first principles.

## I. INTRODUCTION

We consider the frequently occurring situation in which limited duration time series data from some dynamical process is available, but a first-principles-based model of how that data is produced is either unavailable or too inaccurate to be useful. Thus, if one is interested in diagnosing ergodic properties of the underlying processes producing the data, one is restricted to do so based only on the data itself. We call such a method “model-free.” Model-free analysis of dynamical time series is a long-standing subject of study in nonlinear dynamics<sup>5–7</sup>. Perhaps the most wide-spread approach uses delay-coordinate embedding<sup>5–13</sup>. In this article, we discuss a very promising, entirely different approach to model-free analysis of dynamical time series. Our approach is based upon recent significant advances in the area known as *machine learning*. In particular, we will apply a type of machine learning known as reservoir computing<sup>4</sup>, and, for definiteness, we focus on the problem of determining the Lyapunov exponents of the data-generating system. For this application, the key ability we require from machine learning is to replicate the ergodic properties of the system generating the input, and we call this replicating the “climate.”

The rest of this article is organized as follows. Section II reviews reservoir computing and its use for short-term prediction of chaotic time series. Section III illustrates our method using the well-known Lorenz 1963 model<sup>14</sup>, and discusses the ability of reservoir computers to replicate the (long-term) climate. Section IV uses our approach to evaluate the Lyapunov exponents of the Kuramoto-Sivashinsky (KS) equation<sup>15–17</sup> with periodic boundary conditions. This system provides an example of extensive spatiotemporal chaos<sup>18–21</sup>, for which the attractor dimension and number of positive Lyapunov exponents increases linearly with the periodicity length  $L$ . In particular, Sec. IV considers cases with many positive Lyapunov exponents. The paper concludes with further discussion in Sec. V.

The main conclusion of this paper is that our machine learning approach offers a very attractive model-free method for obtaining Lyapunov exponents from data. Particularly notable are our results from Sec. IV where we obtain excellent agreement for all of the positive Lyapunov exponents and many of the negative exponents for a moderately high-dimensional system. In comparison with delay coordinate embedding, we remark that our method appears to be simpler to implement, and does not appear to suffer from the problem of yielding spurious positive Lyapunov exponents (E.g., see [22], [23]). These papers and references therein discuss the mechanism responsible for spurious positive Lyapunov exponents in delay coordinate embedding and how to fix the problem. Since this mechanism is inherently absent in our method, we do not expect, and indeed, have not found spurious positive exponents). More broadly, our paper suggests that machine learning is useful for analysis of data from chaotic systems (e.g., previous work has treated model-free machine learning for prediction of future evolution of the states of a dynamical system<sup>24</sup> and for inference of unmeasured dynamical variables<sup>25</sup>).

## II. RESERVOIR COMPUTERS, SHORT TERM PREDICTION AND ATTRACTOR CLIMATE

Reservoir computers<sup>4</sup> originate from an idea independently put forth about 16 years ago in two papers<sup>26,27</sup>. The general approach is illustrated in Fig. 1(a), which shows an input vector  $\mathbf{u}(t)$  fed into a “reservoir” (labeled R in Fig. 1(a)) through an input-to-reservoir coupler (labeled I/R), with an output vector  $\mathbf{v}$  coupled from the reservoir through an output coupler (labeled R/O). We regard the couplers as acting instantaneously and without memory (i.e., their output depends solely on their current input). Importantly, the reservoir has memory (i.e., it has internal dynamics so its state depends on its history). We assume that it receives input at discrete times  $t$ , and that its input  $\mathbf{W}_{in}\mathbf{u}(t)$  is combined with the reservoir state  $\mathbf{r}(t)$  to produce its output  $\mathbf{r}(t + \Delta t)$ . In general, the reservoir can be any complex dynamical system with many state variables, but here we follow Refs. [26,27] and consider the reservoir to be a large random network with  $D_r$  nodes and an  $D_r \times D_r$  adjacency matrix  $\mathbf{A}$ . Specifically, we will henceforth consider the particular implementation (similar

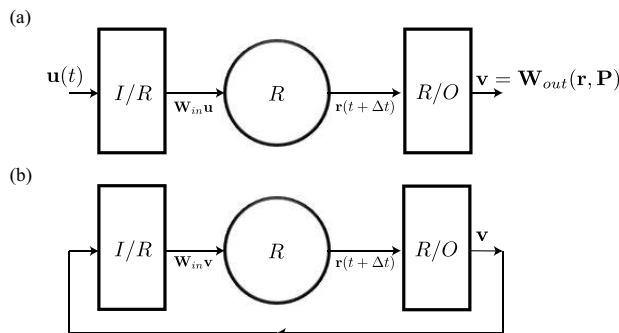


FIG. 1. (a) Configuration of the reservoir in the training phase corresponding to Eqs. 1 and 2. (b) Reservoir configuration in the prediction phase corresponding to Eq. 4.

to Ref. [24]) of Fig. 1(a) given by

$$\mathbf{r}(t + \Delta t) = \tanh[\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{in}\mathbf{u}(t)], \quad (1)$$

$$\mathbf{v}(t + \Delta t) = \mathbf{W}_{out}(\mathbf{r}(t + \Delta t), \mathbf{P}), \quad (2)$$

where  $\mathbf{r}(t)$  represents the scalar states  $r_i(t)$  of the  $D_r$  network reservoir nodes,  $\mathbf{r} = [r_1, r_2, \dots, r_{D_r}]^T$ ; in Eq. (1),  $\mathbf{W}_{in}$  is a  $D_r \times D$  matrix, where  $D$  is the dimension of  $\mathbf{u}$ ; also, in Eq. (1), for a vector  $\mathbf{q} = (q_1, q_2, \dots)^T$  the quantity  $\tanh(\mathbf{q})$  is the vector  $(\tanh(q_1), \tanh(q_2), \dots)^T$ . In Eq. (2),  $\mathbf{W}_{out}$  maps the  $D_r$  dimensional vector  $\mathbf{r}$  to the output  $\mathbf{v}$ , which, for the situations considered in this article, has the same dimension  $D$  as  $\mathbf{u}$ . In addition, we assume that  $\mathbf{W}_{out}$  depends on a large number of adjustable parameters given by the elements of the matrix  $\mathbf{P}$ , and that  $\mathbf{W}_{out}(\mathbf{r}, \mathbf{P})$  depends linearly on  $\mathbf{P}$  (e.g., in past work the choice  $\mathbf{W}_{out}(\mathbf{r}, \mathbf{P}) = \mathbf{P}\mathbf{r}$  has often been used).

In general, the goal of the system in Fig. 1(a) is for the outputs  $\mathbf{v}(t)$  to approximate the desired outputs,  $\mathbf{v}_d(t)$ , appropriate to the inputs  $\mathbf{u}(t)$  (e.g., in a pattern recognition task  $\mathbf{u}(t)$  might represent a sequence of patterns, and  $\mathbf{v}_d(t)$  would represent pattern classifications the patterns). To this end, during a training period,  $-T \leq t \leq 0$ , an input  $\mathbf{u}(t)$  is fed into the reservoir and the resulting reservoir state evolution  $\mathbf{r}(t)$ , along with  $\mathbf{u}(t)$ , are recorded and stored as “training data.” Then the parameters  $\mathbf{P}$  are chosen (“trained”) so as to approximately minimize the mean squared difference between  $\mathbf{v}(t)$  and its desired value  $\mathbf{v}_d(t)$ . As is common in reservoir computing, we use the Tikhonov regularized regression procedure<sup>28</sup> to find an output matrix  $\mathbf{P}$ , that minimizes the following function,

$$\sum_{-T \leq t \leq 0} \|\mathbf{W}_{out}(\mathbf{r}(t), \mathbf{P}) - \mathbf{v}_d(t)\|^2 + \beta \|\mathbf{P}\|^2, \quad (3)$$

where  $\|\mathbf{P}\|^2$  denotes the sum of the squares of elements of  $\mathbf{P}$ . The regularization constant  $\beta > 0$  discourages overfitting by penalizing large values of the fitting parameters (In Sec. IV we used a value  $\beta > 0$ , but for Sec. III we found that using  $\beta = 0$  was sufficient). For a given task, one hopes that for large enough  $D_r$  and  $T$ , the system in Fig. 1(a) will yield subsequent ( $t > 0$ ) outputs  $\mathbf{v}(t)$  that closely approximate the desired  $\mathbf{v}_d(t)$ . Because  $\mathbf{W}_{out}(\mathbf{r}, \mathbf{P})$  is taken to be linear in  $\mathbf{P}$ , the problem of determining the parameters  $\mathbf{P}$  that minimize Eq. (3) is one of linear regression for which there are well-established techniques<sup>29</sup>. This approach has been shown to work extremely well for a wide variety of tasks<sup>4</sup>.

We now consider the task of prediction for the case where  $\mathbf{u}(t)$  depends on the state of some deterministic dynamical system. This problem was originally considered in the reservoir computer framework by Jaeger and Haas<sup>24</sup>. The idea is to take the desired output to be the same as the input,  $\mathbf{v}_d(t + \Delta t) = \mathbf{u}(t + \Delta t)$ . When one wishes to commence

prediction at  $t = 0$ , the configuration is switched from that in Fig. 1(a) to that in Fig. 1(b), and the reservoir system is run autonomously according to the following equation.

$$\mathbf{r}(t + \Delta t) = \tanh [\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{in}\mathbf{W}_{out}(\mathbf{r}(t), \mathbf{P})]. \quad (4)$$

The output of the autonomous reservoir,  $\mathbf{v}(t) = \mathbf{W}_{out}(\mathbf{r}(t), \mathbf{P})$ , gives the predicted value  $\mathbf{u}(t)$  for  $t > 0$ . Jaeger and Haas<sup>24</sup>, using the example of the Lorenz system<sup>14</sup>, indeed verified that this prediction scheme works and gives good short term predictions. As expected, the chaotic amplification of small errors leads to eventual breakdown of the prediction, limiting the prediction time. However, as shown in the next two sections, following this breakdown of short-term prediction, the evolution of  $\mathbf{v}(t)$  often provides an accurate approximation for the climate corresponding to  $\mathbf{u}(t)$ , and can be used in particular to compute Lyapunov exponents of the process that generated  $\mathbf{u}(t)$ .

### III. EXAMPLE 1: THE LORENZ SYSTEM AND THE QUESTION OF WHETHER THE CLIMATE IS REPLICATED

In this section we illustrate the capability of our technique to replicate the “climate” of the Lorenz 1963 system<sup>14</sup>,

$$\begin{aligned} \dot{x} &= 10(y - x), \\ \dot{y} &= x(28 - z) - y, \\ \dot{z} &= xy - 8z/3. \end{aligned} \quad (5)$$

We construct and train reservoir computers with input  $\mathbf{u} = (x, y, z)^T \in \mathbb{R}^3$  and output  $\mathbf{v} \in \mathbb{R}^3$ , following Sec. II. The reservoir network is built from a sparse random Erdős-Rényi network whose average degree is  $d = 6$ . Each non-zero element in the adjacency matrix is drawn independently and uniformly from  $[-a, a]$ , and  $a > 0$  is adjusted so that the spectral radius of  $\mathbf{A}$  (the largest magnitude of its eigenvalues) has a desired value  $\rho$ . During the training phase,  $-T \leq t \leq 0$  (where  $T = 100$ ), the reservoir computer evolves following Eq. (1) with  $\Delta t = 0.02$ . In this Lorenz example, the reservoir output  $\mathbf{v}(t) = \mathbf{W}_{out}(\mathbf{r}(t), \mathbf{P})$  is defined as

$$\mathbf{v}(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1 \mathbf{r}(t) \\ \mathbf{p}_2 \mathbf{r}(t) \\ \mathbf{p}_3 \tilde{\mathbf{r}}(t) \end{bmatrix} \quad (6)$$

where  $\mathbf{p}_1$ ,  $\mathbf{p}_2$ , and  $\mathbf{p}_3$  are the rows of the  $3 \times D_r$  matrix  $\mathbf{P}$ . The quantity  $\tilde{\mathbf{r}}$  in the third line of Eq. (6) is defined in a way such that the first half of its elements are the same as that of  $\mathbf{r}$ , i.e.,  $\tilde{r}_i = r_i$  for half ( $D_r/2$ ) of the reservoir nodes, while  $\tilde{r}_i = r_i^2$  for the remaining half of the reservoir node (Our use here of  $\tilde{\mathbf{r}}(t)$ , rather than  $\mathbf{r}(t)$ , to predict  $z(t)$  is related to the  $x \rightarrow -x$ ,  $y \rightarrow -y$  symmetry of the Lorenz equations as discussed in Ref. [25]).

After we compute  $\mathbf{r}(t)$  for the training period,  $-T \leq t \leq 0$ , we calculate the output weight parameters  $\mathbf{P}$  that minimize the function in Eq. (3) with the desired output being the state variables from the Lorenz system,  $\mathbf{v}_d(t) = [x(t), y(t), z(t)]^T$  (in an actual physical experiment, we assume  $\mathbf{u}(t) = \mathbf{v}_d(t)$  to have been measured for  $-T \leq t \leq 0$ ). After we find the output weights, we evolve the reservoir with the reconfigured reservoir system (Fig. 1(b)).

Following the above described procedure, We now report and compare results for two simulations using reservoir configurations with  $\rho = 1.2$  (denoted R1) and  $\rho = 1.45$  (denoted R2). The prediction for  $0 < t \leq 25$  for both trained reservoirs are shown in Fig. 2(a) (R1 with  $\rho = 1.2$ ) and Fig. 2(b) (R2 with  $\rho = 1.45$ ). Both reservoirs R1 and R2 generate correct short-term predictions and then deviate from the actual Lorenz trajectories, which is expected since any small error grows exponentially due to the chaotic dynamics of the Lorenz system. However, after the failure of the short-term prediction, the two reservoirs

Parameter	Value	Parameter	Value
$D_r$	300	$d$	6
$T$	100	$\Delta t$	0.02
$T/\Delta t$	5000	$\beta$	0
$\rho$	1.2	$\sigma$	0.1

TABLE I. Standard reservoir parameters used for a successful climate replication of the Lorenz system (referred to in the text as the *R1* reservoir). The *R2* reservoir uses the same parameters with a different spectral radius,  $\rho = 1.45$ .

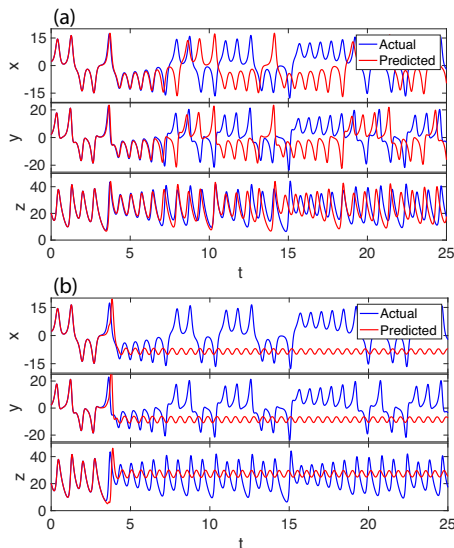


FIG. 2. (a) The state prediction (red) of the R1 reservoir and the actual trajectories (blue) of the Lorenz system for  $0 < t \leq 25$ . The spectral radius of the reservoir is 1.2. (b) The state prediction (red) of the R2 reservoir and the actual trajectories (blue) of the Lorenz system for  $0 < t \leq 25$ . The spectral radius of the reservoir is 1.45.

show qualitatively different dynamical patterns. In Fig. 2(a), it seems that, after  $t \approx 7$ , although the R1 prediction deviates from the actual trajectory, the long-term dynamics appears to resemble that of the original Lorenz system. In contrast, as shown by Fig. 2(b), this is clearly not the case for R2.

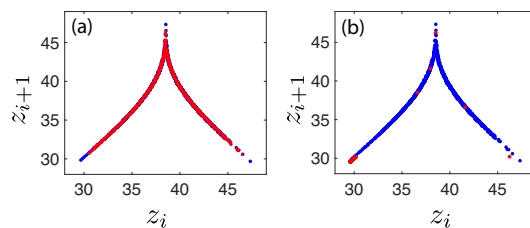


FIG. 3. The return map of the actual and the predicted  $z$ -coordinate of the Lorenz system. This plot is made with time series of length 1000, where the blue dots are from the actual Lorenz system, and the red dots overlaying the blue dots are from the prediction. The left panel shows the return map of the long term prediction of the R1 reservoir with  $\rho = 1.2$ , while the right panel is from the R2 reservoir with  $\rho = 1.45$ .

In Fig. 3 we present a more accurate test than visual inspection of Figs. 2(a) and 2(b) for correctness of the climate. To do this, we follow Lorenz's procedure of plotting the

	Actual Lorenz System	R1 System	R2 System
$\Lambda_1$	0.91	0.90	0.01
$\Lambda_2$	0.00	0.00	-0.1
$\Lambda_3$	-14.6	-10.5	-9.9

TABLE II. Three largest Lyapunov exponents  $\Lambda_1 \geq \Lambda_2 \geq \Lambda_3$  for the Lorenz system (Eq. (5)), and for the reservoir set up in the configuration of Fig. 1(b) for R1 and R2. Since the reservoir system that we employ is a discrete time system, while the Lorenz system is a continuous system, for the purpose of comparison,  $\Lambda_1$ ,  $\Lambda_2$ , and  $\Lambda_3$  are taken to be per unit time; that is, their reservoir values (columns 2 and 3) are equal to the reservoir Lyapunov exponents calculated on a per iterate basis divided by  $\Delta t$ .

return map of successive maxima of  $z(t)$ . We first obtain  $z(t)$  for a long period of time,  $0 < t < 1000$ , for both the actual and the predicted time series. We then locate all local maxima of the actual and predicted  $z(t)$  in time order and denote them  $[z_1, z_2, \dots, z_m]$ . Then, we plot consecutive pairs of those maxima  $[z_i, z_{i+1}]$  for  $i = 1, \dots, m - 1$  as dots in Figs. 3. The blue dots in both panels of Figs. 3 are from the actual Lorenz system, while the red dots printed over the blue dots are from the reservoir output prediction ( $v_3$ ) of  $z(t)$ . As confirmed by Fig. 3(a) the red dots produced by the R1 reservoir continue to fall on top of the blue dots (from the actual Lorenz system) throughout the entire run time ( $0 < t < 1000$ ). In contrast, Fig. 3(b) shows that the blue dots remain largely uncovered, because, as indicated in the third panel of Fig. 2(b), the maximum value of  $z(t)$  for  $t > 5$  is at a fixed point  $z_{max} \approx 30$ . Thus the R1 reservoir very accurately succeeds in reproducing the long-time climate of the attractor, while the R2 reservoir does not, and this is so even though both setups are apparently capable of producing useful short term predictions. (We have also obtained similar results for many other simulations.) Thus some parameter adjustment may be necessary to avoid unsuccessful reproduction of the climate. Fortunately, we usually find that when the climate is not reproduced it is fairly evident (as in Fig. 2(b), as well as Fig. 5 of the next section). More quantitatively, a promising general means of assessing whether the reservoir system has succeeded in mimicking the climate is to first use the training data to obtain finite-time estimates of the system's ergodic properties (e.g., frequency-power spectra, time correlations, moments etc.). Once this is done, one can test whether those estimates are consistent with determinations of the same quantities obtained from the long-term reservoir dynamics. Section IV provides such an assessment for the Kuramoto-Sivashinsky system.

The reservoir in the autonomous configuration of Fig. 1(b) represents a known discrete-time,  $D_r$ -dimensional dynamical system (since we know  $\mathbf{W}_{in}$ ,  $\mathbf{A}$ , and the output parameters  $\mathbf{P}$  determined by the training). We compute the equations for the evolution of the tangent map corresponding to Eq. (4) and evolve a set of  $m$  mutually orthogonal tangent vectors  $\mathbf{R}(t) = \{\delta \mathbf{r}_j\}_{j=1}^m$  along with Eq. (4). We then compute the largest  $m$  Lyapunov exponents of the reservoir dynamical system in the configuration shown in Fig. 1(b) using a standard algorithm based on  $QR$  decomposition (e.g., see Ref. [7]) of the matrix  $\mathbf{R}(t)$ . The two right-most columns of Table II show the three largest Lyapunov exponents,  $\Lambda_1 \geq \Lambda_2 \geq \Lambda_3$ , of the reservoir system in the autonomous configuration, Fig. 1(b), for the R1 reservoir (for which climate reproduction succeeds), and for the R2 reservoir (for which climate reproduction fails).

Comparing the Lyapunov exponents of the Lorenz system (first column of Table II) with those of the R1 reservoir, we see that the largest Lyapunov exponent of the R1 reservoir is a good approximation to the largest Lyapunov exponent of the Lorenz system. Also, consistent with the small value of  $\Delta t$ , the reservoir dynamics approximates that of a flow for which  $\Lambda_2$  should be (and is) approximately zero. On the other hand, we see that the third Lyapunov exponent of the R1 system is less negative than the negative Lyapunov exponent of the true Lorenz system. In contrast with the good agreement of the  $\Lambda_1$  values for the Lorenz system and the R1 reservoir, the positive Lyapunov exponent of the Lorenz system fails to be reproduced by the R2 system whose largest Lyapunov exponent is approximately

zero; this is consistent with the observation from Fig. 2(b) that the long term reservoir attractor for R2 appears to be a periodic orbit.

The significant conclusion from the above is that the R1 system, as a result of successfully reproducing the climate, can be utilized to obtain an approximation to the positive and zero Lyapunov exponents of the process generating its input. We note, however, that the R1 system does not accurately reproduce the true negative Lyapunov exponent of the Lorenz attractor.

The inaccurate reservoir estimation of  $\Lambda_3$ , noted above, can be understood by noting that, although the return map in Fig. 3 appears to be a curve, this apparent “curve” must, as noted by Lorenz<sup>14</sup>, actually have some small width. The R1 reservoir succeeds in *approximating* the attractor of the Lorenz system as reflected by its apparent good reproduction of the return map shown in Fig. 3(a). In order to do this, however, the reservoir need not reproduce the very thin transverse structure within the apparent curve. Since, this very thin structure, as we next discuss, is the primary orbital evidence of the value of  $\Lambda_3$ , one might not expect the reservoir to accurately reproduce this very negative Lyapunov exponent. Specifically, using the Kaplan-Yorke formula for the information dimension<sup>30</sup> of the fractal Lorenz attractor, we obtain a dimension of  $[2 + (\Lambda_1/|\Lambda_3|)] = 2.06$ , corresponding to 1.06 for the dimension of the structure in the return map (Fig. 3(a)). This dimension is very close to one, in agreement with the approximate curve-like character of the return map. However, close examination of the return map “curve” of the Lorenz attractor has previously shown that, within its thickness, there is a fractal set of small transverse dimension (presumably  $\Lambda_1/|\Lambda_3| = 0.06$ ). On the other hand, the Kaplan-Yorke dimension for the return map for the climate of the R1 reservoir attractor is about 1.09. Since the primary orbital difference reflected by differing values of  $\Lambda_3$  is the difference in very thin structure features of the return map that have only a small effect on the climate dynamics, it is not surprising that the R1 reservoir, while giving a good approximation to the true climate of the Lorenz system, gives only a rough approximation of  $\Lambda_3$ .

#### IV. EXAMPLE 2: THE TASK OF DETERMINING A LARGE NUMBER OF LYAPUNOV EXPONENTS OF A HIGH DIMENSIONAL SPATIOTEMPORAL CHAOTIC SYSTEM FROM DATA

We now consider a modified version of the Kuramoto-Sivashinsky (KS) system defined by the partial differential equation for the function  $y(x, t)$

$$y_t = -yy_x - \left[ 1 + \mu \cos \left( \frac{2\pi x}{\lambda} \right) \right] y_{xx} - y_{xxx}, \quad (7)$$

in the region  $0 \leq x < L$  with periodic boundary conditions,  $y(x, t) = y(x + L, t)$ , and  $\lambda$  chosen so that  $L$  is an integer multiple of  $\lambda$ . This equation reduces to the standard KS equation when  $\mu = 0$ . The cosine term makes the equation spatially inhomogeneous. We will subsequently consider the cases  $\mu = 0$  and  $\mu \neq 0$  in order to discuss the effect of the symmetries of the KS equation on the learning dynamics of the reservoir computer.

By numerically integrating Eq. (7) on an evenly spaced one-dimensional grid of size  $Q$ , we obtain a discretized multivariate data set of  $Q$  time series,

$$\mathbf{u}(t) = [y(\Delta x, t), y(2\Delta x, t), \dots, y(Q\Delta x, t)]^T, \quad (8)$$

$$\Delta x = L/Q.$$

As in the case of the Lorenz equations discussed in Sec. III, we consider the situation where we have access to the time series data but do not have information about the dynamical equation that generated the time series. In the absence of a model, we will use the data to train a reservoir computer to emulate the behavior of the true dynamical system, in this case Eq. (7).

The reservoir network is as described in Sec. II with the parameters listed in Table III. In the training phase, Fig. 1(a), we evolve the reservoir according to Eq. (1) from  $t = -T$



Parameter	Value	Parameter	Value
$D_r$	9000	$d$	3
$T$	20000	$\Delta t$	0.25
$T/\Delta t$	80000	$\beta$	0.0001
$\rho$	0.4	$\sigma$	0.5

TABLE III. Reservoir parameters used for the successful replication of the climate of the Kuramoto-Sivashinsky system shown in Fig. 4.

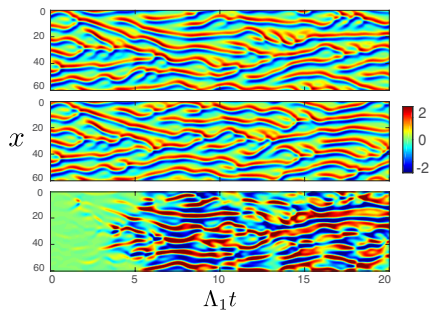


FIG. 4. Top panel: True state,  $y(x, t)$ , of the standard KS system after  $t = 0$ . Middle panel: Reservoir prediction. Bottom panel: Difference between the true state and the reservoir prediction. The parameters of the KS equation are  $L = 60$ ,  $\mu = 0$ .  $\Lambda_1$  denotes the largest Lyapunov exponent.

to  $t = 0$ . Next, we use Tikhonov regularized regression (see Eq. (3)) to compute the output parameters,  $\mathbf{P}$  such that  $\mathbf{W}_{out}(\mathbf{r}, \mathbf{P}) = \mathbf{P}\tilde{\mathbf{r}}(t) \simeq \mathbf{u}(t)$  for  $-T \leq t < 0$ . Here  $\tilde{\mathbf{r}}$  is a  $D_r$ -dimensional vector such that the  $i^{\text{th}}$  component of  $\tilde{\mathbf{r}}$  is  $\tilde{r}_i = r_i$  for half the reservoir nodes and  $\tilde{r}_i = r_i^2$  for the remaining half. With the output parameters determined, we let the reservoir evolve autonomously for  $t > 0$  as shown in Fig. 1(b) according to Eq. (4).

The predictions made by the reservoir system for  $t > 0$  are given by,  $\mathbf{W}_{out}(\mathbf{r}(t), \mathbf{P})$ . Figure 4 shows the time evolution of one such reservoir prediction for  $t > 0$  (middle panel), along with the true state (top panel) of the KS equation and the deviation (bottom panel) of the reservoir prediction from the true state (i.e., the difference between the top panel and the middle panel). Note that in Fig. 4 time (the horizontal axis) is in units of the Lyapunov time ( $\Lambda_1^{-1}$ , where  $\Lambda_1$  is the largest Lyapunov exponent of the KS attractor). We see that the reservoir gives good short term prediction for about 5 multiples of the Lyapunov time. A visual inspection of Fig. 4 suggests that the reservoir prediction may have also learned the correct ‘climate’ of the KS system even after the state of the reservoir dynamical system has diverged from the true state of the KS system.

Figure 5 shows an example of an alternate scenario for another set of the reservoir parameters ( $\rho = 3.1$ ,  $D_r = 5000$  with the rest of the parameters as shown in Table III). In this case, the reservoir still predicts accurately for a short period of time. However, the long term climate of the signal generated by the reservoir is no longer similar to that of the true KS climate.

A more quantitative assessment of the climate reproduction can be obtained by calculating the power spectrum of the reservoir prediction and comparing it with the power spectrum of the training data. Figure 6 shows the power spectrum of the training data, along with the power spectrum of the dynamics of the autonomous reservoir system in Figs. 4 and 5. We see that the reservoir system corresponding to Fig. 4 succeeds in reproducing the training data power spectrum, thus indicating that the long term system orbit reproduces the climate of the training data. On the other hand, the power spectrum of the reservoir system corresponding to Fig. 5 confirms our visual assessment that this reservoir system fails to reproduce the climate of the training data.

Similar to what was done in Sec. III, we use our complete knowledge of the dynamics of



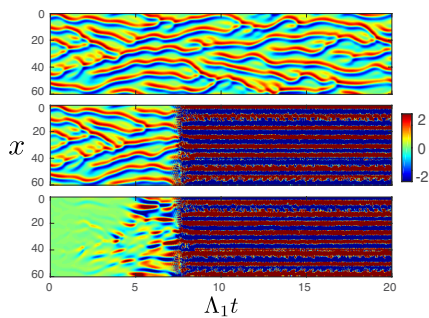


FIG. 5. Top panel: True state,  $y(x, t)$ , of the standard KS system after  $t = 0$ . Middle panel: Reservoir prediction with a reservoir of size  $D_r = 5000$  and  $\rho = 3.1$ . The rest of the parameters are as given in Table III. Bottom panel: Difference between the reservoir prediction and the true KS state. We see that in this case, the reservoir gives us an accurate short term prediction (i.e., the ‘weather’) but the long term ‘climate’ of the autonomous reservoir dynamical system does not resemble the climate of the true KS system for this poorly chosen set of parameters.  $\Lambda_1$  denotes the largest Lyapunov exponent.

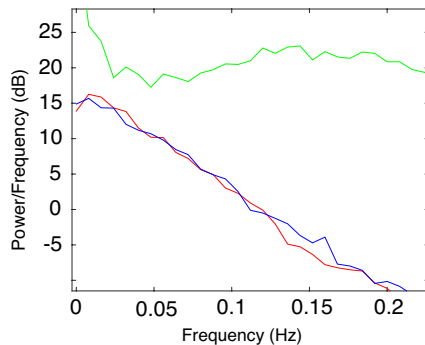


FIG. 6. Power spectrum of the KS training data (blue), of the reservoir prediction with the same parameters as in Fig. 4 (red), and of the reservoir prediction with parameters as in Fig. 5 (green). All power spectra have been computed at a single spatial gridpoint from a time series of length  $15000 \Delta t$  time steps. The power spectra are smoothed by dividing a time series into 30 intervals, computing the power spectrum of each interval and then averaging over all the intervals.

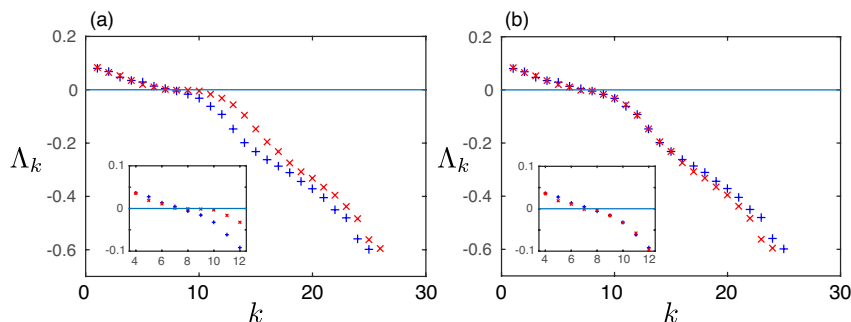


FIG. 7. (a) Estimating the Lyapunov exponents of the homogeneous ( $\mu = 0$ ) KS equation. First 26 Lyapunov exponents of the trained reservoir dynamical system running in autonomous prediction mode (blue ‘+’ markers) and the standard (i.e.,  $\mu = 0$ ) KS system (red ‘x’ markers). The parameters of Eq. (7) are  $L = 60$ ,  $\mu = 0$ . (b) The same plot as (a), except, the two near-zero exponents of the KS system ( $\Lambda_7$  and  $\Lambda_8$ ) are removed from the spectrum. Inset: a close up of the spectra around the zero crossing. All Lyapunov exponents in this figure and Fig. 8 were computed from a trajectory of length  $10000 \Delta t$  time steps, which we found to be sufficiently long for convergence.

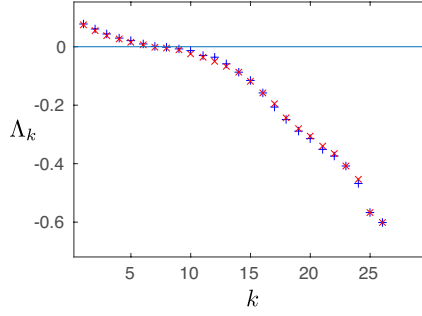


FIG. 8. Estimating the Lyapunov exponents of the inhomogeneous ( $\mu > 0$ ) KS equation. First 26 Lyapunov exponents of the trained reservoir dynamical system running in autonomous prediction mode (blue '+' markers) and the modified (i.e.,  $\mu > 0$ ) KS system (red 'x' markers). The parameters of Eq. (7) are  $L = 60$ ,  $\mu = 0.1$  and  $\lambda = 15$ .

the reservoir computer to evaluate its Lyapunov exponents. By independently evaluating the Lyapunov exponents directly from the KS equation, Eq. (7), we obtain the true Lyapunov exponents and compare them with the corresponding Lyapunov exponents of the reservoir dynamical system.

Figure 7(a) shows the Lyapunov spectrum of the standard ( $\mu = 0$ ) KS system with  $L = 60$  (red 'x' markers), where, by definition the subscript  $k$  is such that  $\Lambda_k \geq \Lambda_{k+1}$ . The Lyapunov exponents of the reservoir trained to emulate this system are shown on the same axes (blue '+' markers). We observe that the positive Lyapunov exponents of the reservoir system match the corresponding exponents of the KS system very well. However, the negative exponents of the two systems do not seem to agree with each other at first glance. We argue below that the standard KS system has three zero Lyapunov exponents, and we posit that the reservoir is unable to reproduce two of them. Indeed, Fig. 7(b) shows that if we remove the two of the computed exponents closest to zero ( $\Lambda_7$  and  $\Lambda_8$ ) for the KS system, the negative Lyapunov exponents of the reservoir system match those of the KS system very well.

We show now that when  $\mu = 0$  (as for Fig. 7), the standard KS equation (7) has three zero Lyapunov exponents associated with three continuous symmetries, namely time-translation invariance, space-translation invariance and the so-called Galilean invariance. Time and space translation invariance imply that if  $y(x, t)$  is a solution, then so are  $y(x, t + t_0)$  and  $y(x + x_0, t)$ . By Galilean invariance, we mean that for every solution  $y(x, t)$  of the KS equation and an arbitrary constant  $v$ ,  $y(x - vt, t) + v$  is also a solution. This can be verified by direct substitution in Eq. (7) with  $\mu = 0$ . Replacing  $t_0$ ,  $x_0$ , and  $v$  by differentials ( $t_0 \rightarrow \delta t_0$ ,  $x_0 \rightarrow \delta x_0$ ,  $v \rightarrow \delta v$ ), we have that,  $\delta y(x, t) = \frac{\partial y(x, t)}{\partial t} \delta t_0$ ,  $\delta y(x, t) = \frac{\partial y(x, t)}{\partial x} \delta x_0$  and  $\delta y(x, t) = \left[1 - t \frac{\partial y(x, t)}{\partial x}\right] \delta v$  all represent perturbations,  $y(x, t) + \delta y(x, t)$ , of Eq. (7) that are, to linear order in the differentials, solutions of Eq. (7). That is, all three of these  $\delta y(x, t)$  are solutions of the variational equation,  $\delta y_t + \delta y y_x + y \delta y_x + \delta y_{xx} + \delta y_{xxx} = 0$ . Furthermore, since the original solution  $y(x, t)$  does not decay exponentially to zero, nor increase exponentially to infinity, we conclude that these three expressions for  $\delta y$  represent Lyapunov vectors with zero Lyapunov exponents.

To see why the reservoir does not reproduce the Galilean symmetry-associated zero Lyapunov exponent in the  $\mu = 0$  case, notice that there is a corresponding conserved quantity  $c = \int y(x, t) dx$ . A particular KS system trajectory in phase space is thus restricted to a hypersurface with a constant value of  $c$  (say,  $c = c_0$ ). Since the reservoir is trained with data from a single trajectory, it does not learn the dynamics of perturbations that take the trajectory off the  $c_0$  hypersurface. We are not certain why the reservoir does not reproduce both of the other two zero exponents.

As a further example that does not have additional symmetries beyond time-translation, we consider (Fig. 8) a KS equation with a nonzero value of  $\mu$  ( $L = 60, \lambda = 15, \mu = 0.1$ ).

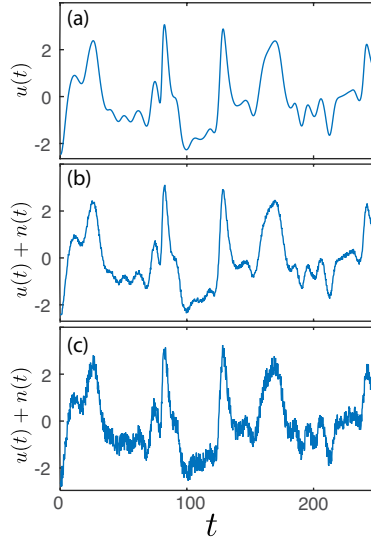


FIG. 9. (a) Single scalar component  $u(t)$  of the time series  $\mathbf{u}(t)$  generated from the KS system (Eq. (7)) with  $L = 60$ ,  $\lambda = 15$  and  $\mu = 0.1$ . The time series in (a) with added noise,  $u(t) + n(t)$ , of noise strengths  $f = 0.05$  and  $f = 0.2$  are shown in (b) and (c) respectively.

As before, we train the reservoir using the time series data from the symmetry broken KS equation. After training, we run the reservoir in autonomous prediction mode (Fig. 1(b)) and calculate its Lyapunov spectrum. Figure 8 shows that the reservoir reproduces the Lyapunov spectrum of the true KS system accurately in this case. Notably, in contrast with the case  $\mu = 0$ , this good agreement is obtained without the need of discarding two zero Lyapunov exponents. We continue to use this modified KS system in the experiments described below. For the cases shown in Figs. 7(b) and 8, the information dimension of the attractor, as computed from the Kaplan-Yorke conjecture [30], is about  $D_{KY} \approx 15$  (roughly, the value of  $k$  at which  $\sum_{j=1}^k \Lambda_j$  first becomes negative). We see from Fig. 7(b) and Fig. 8 that the reservoir continues to give reasonable estimates of  $\Lambda_k$  even for  $k > D_{KY}$ . This was somewhat surprising to us, especially in view of the inaccurate reservoir estimate of  $\Lambda_3$  in Sec. III.

We now consider the effect of additive measurement noise on our Lyapunov exponent calculation scheme. We simulate measurement noise by adding a random vector  $\mathbf{n}(t)$  to the training data set  $\mathbf{u}(t)$  for all values of  $t$ . That is, at every time step  $\Delta t$ , we replace  $\mathbf{u}$  in Eq. (1) by  $\mathbf{u} + \mathbf{n}$ , and we replace  $\mathbf{v}_d = \mathbf{u}$  used in Eq. (3) by  $\mathbf{v}_d = \mathbf{u} + \mathbf{n}$ . The scalar elements  $n_j(t)$  of the vector  $\mathbf{n}(t)$ , for each value of  $j$  and  $t$ , are independent, identically distributed uniform random variables in the interval  $[-\alpha, \alpha]$ . The constant  $\alpha$  is chosen so that the RMS value of the noise is  $f$  times the RMS value of the noise-free signal  $\mathbf{u}(t)$ . Figure 9(a) shows the noise-free time series at a single grid point, while Figs. 9(b) and 9(c) show the same time series with added noise of strength  $f = 0.05$  and  $f = 0.2$ , respectively. We calculate the Lyapunov exponents of the reservoir as described above. Figure 10 shows the Lyapunov spectrum when the noise level  $f$  is varied from 0.05 to 0.20 along with the true Lyapunov spectrum of the KS equation. We see that the reservoir results for the positive Lyapunov exponents are quite robust to noise for  $f \leq 0.2$ , but that the negative exponents are increasingly depressed to more negative values as  $f$  increases.

We find that the amount of data used to train the reservoir computer can significantly affect the accuracy of the Lyapunov spectrum. The negative Lyapunov exponents are more sensitive than the positive exponents to errors due to insufficient training data. Figure 11 demonstrates this result through a plot of the Lyapunov spectrum of the reservoir trained on varying lengths of data from Eq. (7) with parameters  $L = 60$ ,  $\lambda = 15$  and  $\mu = 0.1$ . In this example we find that we need a training time series of greater than 20000 time steps in

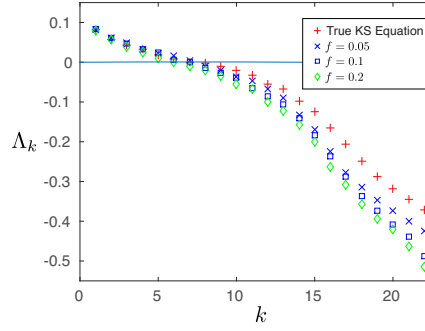


FIG. 10. Lyapunov exponents of the reservoir trained on noisy data from the KS system ( $L = 60$ ,  $\lambda = 15$ ,  $\mu = 0.1$ ). The strength of the noise added to the training data is indicated in the legend.

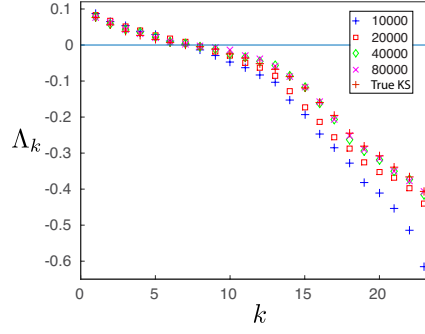


FIG. 11. The Lyapunov spectrum of the reservoir trained using varying lengths of training data from Eq. (7) with parameters  $L = 60$ ,  $\lambda = 15$  and  $\mu = 0.1$ . The legend indicates the length of the training time series in number of  $\Delta t$  steps (i.e.,  $T/\Delta t$ ). For a comparison with a natural time scale of the KS system, we note that 10000  $\Delta t$  time steps equals approximately 200 Lyapunov times.

order to obtain a reasonably accurate estimate of the negative Lyapunov exponents (20000 time steps equals about 400 multiples of the Lyapunov time ( $\Lambda_1^{-1}$ ) which can be considered to be a natural time scale of the KS system).

## V. DISCUSSION AND CONCLUSION

We conclude that a suitably trained reservoir computing system is capable of approximating the ergodic properties of the true system that it was trained on. Remarkably, as shown in Sec. IV, it is possible to use the trained reservoir to calculate a large number of positive and negative Lyapunov exponents of a high dimensional spatio-temporal chaotic system with good accuracy. In the case of the Lorenz equations, our method is successful in calculating the positive and zero Lyapunov exponents with good accuracy. The negative Lyapunov exponent of the true Lorenz system has a high magnitude, and our method is not as successful in accurately calculating the numerical value of this exponent, although it does successfully capture that its magnitude is substantially larger than that of the positive exponent. From a more general point of view, our paper suggests that the development of machine learning techniques for model-free analysis of measured data from chaotic systems may be a fruitful subject for further research.

## VI. ACKNOWLEDGMENT

This work was supported by grant W911NF1210101 from the Army Research Office. We wish to acknowledge useful conversations with Daniel Gauthier, Alexander Hartemink and Roger Brockett.

- <sup>1</sup>G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, IEEE Signal Processing Magazine **29**, 82 (2012).
- <sup>2</sup>I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
- <sup>3</sup>D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, Nature **529**, 484 (2016).
- <sup>4</sup>M. Lukosevicius and H. Jaeger, Computer Science Review **3**, 127 (2009).
- <sup>5</sup>H. Kantz and T. Schreiber, *Nonlinear time series analysis*, Vol. 7 (Cambridge University Press, 2004).
- <sup>6</sup>E. Ott, T. Sauer, and J. A. Yorke, Wiley Series in Nonlinear Science, New York: John Wiley (1994).
- <sup>7</sup>H. Abarbanel, *Analysis of observed chaotic data* (Springer Science & Business Media, 2012).
- <sup>8</sup>F. Takens, by DA Rand and L.-S. Young Springer, Berlin **898**, 366 (1981).
- <sup>9</sup>T. Sauer, J. A. Yorke, and M. Casdagli, Journal of Statistical Physics **65**, 579 (1991).
- <sup>10</sup>D. S. Broomhead and G. P. King, Physica D: Nonlinear Phenomena **20**, 217 (1986).
- <sup>11</sup>A. Brandstater and H. L. Swinney, Physical Review A **35**, 2207 (1987).
- <sup>12</sup>J.-P. Eckmann, S. O. Kamphorst, D. Ruelle, and S. Ciliberto, Physical Review A **34**, 4971 (1986).
- <sup>13</sup>V. Petrov, V. Gaspar, J. Masere, and K. Showalter, Nature **361**, 240 (1993).
- <sup>14</sup>E. N. Lorenz, Journal of the Atmospheric Sciences **20**, 130 (1963).
- <sup>15</sup>B. I. Cohen, J. Krommes, W. Tang, and M. Rosenbluth, Nuclear Fusion **16**, 971 (1976).
- <sup>16</sup>Y. Kuramoto and T. Tsuzuki, Progress of Theoretical Physics **55**, 356 (1976).
- <sup>17</sup>G. Sivashinsky, Physica D: Nonlinear Phenomena **4**, 227 (1982).
- <sup>18</sup>M. C. Cross and P. C. Hohenberg, Reviews of Modern Physics **65**, 851 (1993).
- <sup>19</sup>R. Livi, A. Politi, and S. Ruffo, Journal of Physics A: Mathematical and General **19**, 2033 (1986).
- <sup>20</sup>D. A. Egolf and H. S. Greenside, Nature **369**, 129 (1994).
- <sup>21</sup>A. Pikovsky and A. Politi, Nonlinearity **11**, 1049 (1998).
- <sup>22</sup>H. Kantz, G. Radons, and H. Yang, Journal of Physics A: Mathematical and Theoretical **46**, 254009 (2013).
- <sup>23</sup>T. D. Sauer, J. A. Tempkin, and J. A. Yorke, Physical Review Letters **81**, 4341 (1998).
- <sup>24</sup>H. Jaeger and H. Haas, Science **304**, 78 (2004).
- <sup>25</sup>Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, Chaos: An Interdisciplinary Journal of Nonlinear Science **27**, 041102 (2017).
- <sup>26</sup>H. Jaeger, Bonn, Germany: German National Research Center for Information Technology GMD Technical Report **148**, 13 (2001).
- <sup>27</sup>W. Maass, T. Natschlager, and H. Markram, Neural Computation **14**, 2531 (2002).
- <sup>28</sup>N. Tikhonov, Andreï, V. I. Arsenin, and F. John, *Solutions of ill-posed problems*, Vol. 14 (Winston Washington, DC, 1977).
- <sup>29</sup>X. Yan and X. Su, *Linear regression analysis: theory and computing* (World Scientific, 2009).
- <sup>30</sup>J. L. Kaplan and J. A. Yorke, in *Functional Differential Equations and Approximation of Fixed Points* (Springer, 1979) pp. 204–227.