

Constructive Role of Noise for High-Quality Replication of Chaotic Attractor Dynamics Using a Hardware-Based Reservoir Computer

Irene Estébanez, Ingo Fischer, and Miguel C. Soriano ^{*}

Instituto de Física Interdisciplinar y Sistemas Complejos, CSIC-UIB, Campus Universitat Illes Balears, 07112 Palma de Mallorca, Spain

 (Received 29 April 2019; revised manuscript received 29 August 2019; published 27 September 2019)

Hardware-implemented reservoir computing (RC) has been gaining considerable interest in recent years, in particular for classification and nonlinear-prediction tasks. Such RC systems often perform analog computation and, therefore, may be more sensitive to noise than digital systems; noise has been found to often degrade the computational performance. In contrast, here we demonstrate that noise can also play a constructive role in hardware-based RC. Using a hybrid delay-based RC system with an analog part (non-linearity) and a digital part, we show that the replication of chaotic attractor dynamics is overall improved when the reservoir is trained with an input signal modified by additive Gaussian noise. To quantify the performance of the attractor replication, we suggest two different methods based on recurrence plots and power spectra.

DOI: [10.1103/PhysRevApplied.12.034058](https://doi.org/10.1103/PhysRevApplied.12.034058)

I. INTRODUCTION

In recent years, analog hardware implementations of cognitive computing and neuro-inspired information-processing techniques have been gaining considerable interest. The hardware platforms that have been studied include electronic systems [1], magnetic spin systems [2], MEMS [3] and optical systems [4], among others. The associated hope is to realize versatile, hardware-adapted, fast, energy-efficient systems for applications comprising classification tasks, nonlinear prediction, and system control. Given the huge success of digital computing systems in the last few decades, the move toward analog computing systems might appear surprising, since it brings back all the challenges to deal with noise. However, despite its detrimental role in general, noise can sometimes play an important constructive role in dynamical systems. Noise was shown to be able to enhance desired system properties due to phenomena such as stochastic and coherence resonance [5–7] and it has been used, for example, in stochastic optimization techniques. Here we explore the noise-induced performance enhancement in autonomous attractor replication of a chaotic dynamical system using a hardware-implemented reservoir-computing (RC) system. Autonomous replication of chaotic systems has been demonstrated both experimentally [8,9] and by numerical simulations [10]. However, it has been observed that such autonomous reservoir computers can settle on a

dynamical behavior different from the one they have been trained for [9,10]. We go beyond previous approaches by demonstrating that noise added to the input data can result in better overall replication of the desired chaotic dynamics and enhance robustness against parameter variations.

Analogously to weather and climate, short-term chaotic time-series prediction has been referred to as “weather prediction” of the chaotic dynamics and attractor replication has been referred to as “climate prediction” [11]. While short-term chaotic time-series prediction was considered in the early work on RC by Jaeger and Haas [12], chaotic attractor replication has recently attracted significant attention [8,11]. The short-term prediction of a chaotic system is fundamentally limited by the exponential divergence of close trajectories, characterized by the maximum Lyapunov exponent, and requires precise models, high numerical accuracy, and a large number of data points [13]. In contrast, the reproduction of the long-term behavior of a chaotic system entails capturing the topological properties of the chaotic attractor, not the precise trajectories [10]. In this article, we focus on the experimental replication of the long-term dynamical characteristics (climatelike prediction) of chaotic dynamics with high fidelity.

II. RESERVOIR COMPUTING

A. Basic principles

Our experimental approach builds on the delay-based RC architecture originally introduced by Appeltant *et al.* [1] in 2011. Delay-based RC systems fulfill the

^{*}miguel@ifisc.uib-csic.es

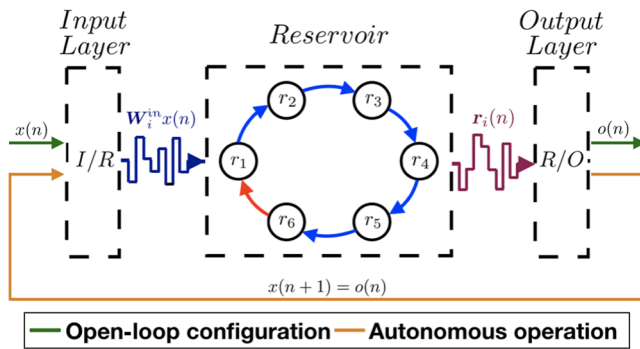


FIG. 1. The one-step-ahead prediction (in green) and the autonomous operation of a delay-based reservoir computer (in orange). In the open-loop configuration the input signal is the original one $x(n)$, while in the autonomous run we close the loop, replacing the original signal by the output signal of the reservoir computer, $o(n-1)$. Node connections are indicated by blue arrows (one step before in time) and a red arrow (two steps back in time). I stands for Input Layer, R for Reservoir and O for Output Layer.

requirements of RC for high-dimensionality and finite fading memory. Instead of requiring a recurrent network with a large number of neurons, here we need only one single nonlinear node and a delayed feedback loop. As illustrated in Fig. 1, the delay-based RC emulates a ring topology. Because of its simplicity, delay-based RC has attracted much attention, resulting in different hardware implementations in electronics [1,14], optoelectronics [15–17], and optics [4,18]. Although the physical substrate of the reservoir may change, the main structure of RC with three layers is preserved: an input layer, the reservoir, and an output layer. Through the input layer, the input signals are fed into the reservoir using weighted connections. These input weights are drawn from a Gaussian distribution and left unaltered. The reservoir exhibits transient nonlinear responses while the input signals are injected. Later, these responses are read out at the output layer via a linear weighted sum of the individual node states. The read-out weights are trained via linear regression. Two possible configurations of the RC system are considered. The open-loop operation (green line in Fig. 1) is used for training and testing stages of a one-step-ahead prediction. During the training stage we calculate the weights, while in the testing stage we evaluate the short-term prediction. The second configuration is obtained by closing the loop with feedback from the output layer as illustrated by the orange line in Fig. 1. This reinjection of the output allows the autonomous generation of complex dynamical behavior.

B. Training on the Rössler system

To illustrate the constructive role of noise in RC, we first aim at the autonomous replication of the x variable of the

Rössler system in the chaotic regime

$$\begin{aligned}\dot{x} &= -y - x, \\ \dot{y} &= x + ay, \\ \dot{z} &= bx - cz + xz,\end{aligned}\quad (1)$$

for the typical values $a = 0.2$, $b = 0.2$, and $c = 5.7$. For the numerical simulation of Eq. (1), we use a fifth-order Runge-Kutta integration method with an integration step of 0.1. Then we subsample the data with a step of 0.3 such that 20 points per natural oscillation of the Rössler dynamics are finally kept. The resulting time series is normalized with zero mean and unit standard deviation. In Fig. 2 we illustrate an example of the original time series (orange line) together with the one predicted by our RC system (dotted blue line). The first 20 samples (shaded region) correspond to the one-step-ahead prediction after the system has been trained in the open-loop operation. The following samples correspond to the autonomous operation when the output feedback loop is enabled. As expected for chaotic dynamics, the output of the autonomous operating system diverges in time from the original simulated trajectory. Still, the time series experimentally generated by the RC system visually resembles the typical oscillations of the original one. In Sec. III, we provide a more-quantitative analysis of this resemblance.

C. Experimental implementation

The experimental setup used for time-series prediction and replication is mainly composed of two elements: an analog electronic circuit and a digital Raspberry Pi board, as shown in Fig. 3. The single nonlinear node of the reservoir is an analog electronic implementation of the Mackey-Glass nonlinearity. The delay is implemented digitally. The experimental setup is detailed in Appendix A.

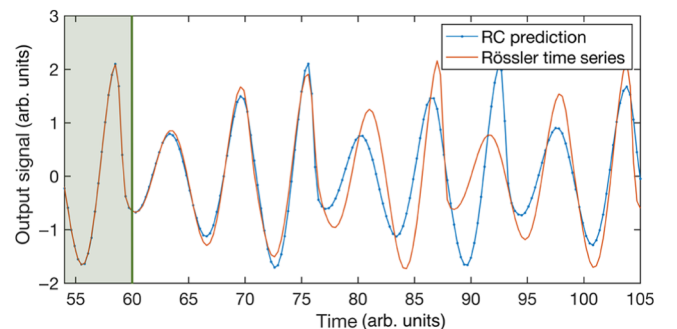


FIG. 2. Example of the time series generated with the delay-based RC. Here we show the performance of the one-step-ahead prediction (shaded region) and the autonomous operation (unshaded region). Experimental results are plotted (dotted blue line) together with the original Rössler time series (orange line).

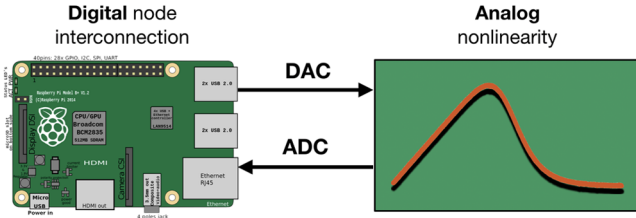


FIG. 3. The RC system based on a single Mackey-Glass nonlinear element with delay in a mixed analog and digital implementation. ADC; analog-to-digital conversion; DAC, digital-to-analog conversion.

For the open-loop case, Fig. 1 (green), we update the delay-based reservoir according to the following equations [19]:

$$\begin{aligned} r_i(n) &= F[\gamma W_i^{\text{in}} x(n) + \beta r_{i-1}(n-1) + \Phi], \quad i=2, \dots, N, \\ r_1(n) &= F[\gamma W_1^{\text{in}} x(n) + \beta r_N(n-2) + \Phi], \\ o(n) &= W^{\text{out}} r(n), \end{aligned} \quad (2)$$

where $r(n)$ and $o(n)$ are the nodes' states of the reservoir and the predicted output values, respectively; W^{in} are the random input connection weights that map the input into the reservoir, which are drawn from a Gaussian distribution; W^{out} are the output weights of the linear combiner that are computed by minimizing the squared error between the target $y(n)$ and the predicted $o(n)$ output values; γ and β are input and feedback scaling factors, respectively; and the added constant Φ allows us to control the operating point within the Mackey-Glass nonlinearity, which is represented in Eq. (2) as F .

D. Reservoir optimization

We proceed to identify the best choice of the parameters (β, γ, Φ) and reservoir size N for the replication of the x variable of the Rössler system. We start from the assumption that the lower the value of the error during the short-term prediction, the better the performance for autonomous operation. The estimation of the prediction error in the experiments is based on an input signal of 4000 samples, divided in 3600 samples for training and 400 samples for testing. From the analysis of the normalized root-mean-square error (NRMSE) in the testing stage, we find the optimal size to be $N = 700$ (Fig. 9 in Appendix B) and the optimal operating point for short-term prediction to be $(\beta, \gamma, \Phi) = (0.3, 200, 600)$ (Fig. 10 in Appendix C). Using these optimized values, we proceed to the climate-like reconstruction of the dynamics using the closed-loop configuration in Fig. 1 (orange) and the output weights computed for the short-term prediction. The update of the reservoir is then realized according to Eq. (2) but with the input value $x(n)$ replaced by the output value predicted one

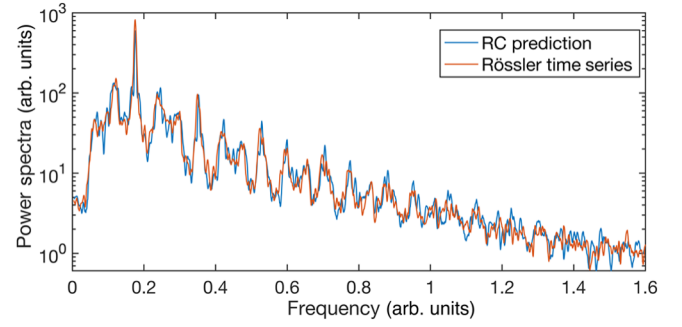


FIG. 4. Frequency spectra of the x variable of the Rössler chaotic attractor. The orange line corresponds to the original time series and the blue line corresponds to the autonomously generated time series for $\beta = 0.3$, $\gamma = 200$, and $\Phi = 600$. The spectra are smoothed with a moving average of five samples for the posterior power-spectrum analysis.

step before $o(n-1)$. To illustrate how the system captures the long-term dynamics, we plot in Fig. 4 the frequency spectra of the time series of the original (orange) and the replica (blue) x variable of the Rössler system.

III. MAIN RESULTS

A qualitative comparison between the original chaotic dynamics and its experimentally generated replica was done previously [8,9]. Here we go one step further and quantify the performance of our system for the long-term reconstruction of the dynamics. For this purpose, we propose two different measures based on recurrence quantification analysis (RQA) [20] and power-spectrum analysis, respectively. Moreover, we present an alternative way to train the RC system that yields a more-robust autonomous generation.

Recurrence plots offer a way to visualize sequential data in a two-dimensional plot [20]. They reveal the times at which a phase-space trajectory visits the same area in phase space. This leads to different structures emerging in the recurrence plots that can be quantified via RQA. Here we compare measures of the RQA in the original time series and in the replica time series. To that end, we compute what we call “divergence” as the sum of the absolute weighted differences of the RQA quantifiers:

$$\begin{aligned} D &= \left| \frac{R(o) - R(p)}{R(o)} \right| + \left| \frac{T(o) - T(p)}{T(o)} \right| \\ &+ \left| \frac{L(o) - L(p)}{L(o)} \right| + \left| \frac{H(o) - H(p)}{H(o)} \right|, \end{aligned} \quad (3)$$

where o stands for “original,” p stands for “predicted,” and the quantifiers recurrence rate (R), determinism (T), averaged diagonal-line length (L), and Shannon entropy (H) are defined in Appendix D.

TABLE I. Different operating points considered in the time-series reconstruction and the corresponding error for the one-step-ahead prediction. The colors refer to Fig. 5.

Operating point (β, γ, Φ)	NRMSE	Color
(0.4, 350, 650)	0.0226 ± 0.0018	Blue
(0.4, 60, 575)	0.0168 ± 0.0010	Red
(0.3, 200, 600)	0.0114 ± 0.0009	Yellow

For the power-spectrum analysis, in turn, we compute the cross-correlation coefficient of the replica and the original spectra on semilogarithmic scale as presented in Fig. 4. Performing the comparison on a semilogarithmic scale enhances the influence of the frequency components besides the dominant frequency peak.

Considering both quantitative methods, we explore the constructive role of noise by training the reservoir with an input signal altered with noise. The input signal is injected into the reservoir according to Eq. (4):

$$x = x_{\text{Rössler}} + \frac{p}{100} \times r, \quad (4)$$

where $x_{\text{Rössler}}$ is normalized with zero mean and unit standard deviation, p is the percentage of noise, and r is a random scalar drawn from the standard normal distribution. Our study of how training with input noise affects the autonomous operation is performed at three different operating points, namely, the optimum point and two neighboring points in the parameter space (see Fig. 10 in Appendix C), that yield different NRMSEs during the one-step-ahead prediction task, as shown in Table I.

For each noise amplitude and parameter set, we first train the system for one-step-ahead prediction to compute the corresponding output weights. These output weights are then loaded into the Raspberry Pi so that we can generate the predicted output online and proceed to test the autonomous operation. After an initialization of 200 samples in an open-loop configuration, allowing the system to settle on the original trajectory, we switch to closed-loop operation with output feedback. Then we let the system evolve autonomously for the next 3800 time steps. Figure 5 shows the divergence and the maximum of the spectral cross-correlation coefficient obtained from the comparison of the autonomously generated time series and the original time series. In Figs. 5(a) and 5(b), the error bars are computed from eight repetitions for each amplitude of the Gaussian noise.

The results in Figs. 5(a) and 5(b) show that the climate-like reconstruction of the dynamics gets on average better (i.e., low divergence and high spectral correlation) when the reservoir is trained with an input signal altered with Gaussian noise with an amplitude of approximately 3%. When we concentrate on the improvement for the optimal operating point alone (yellow line), it might not appear

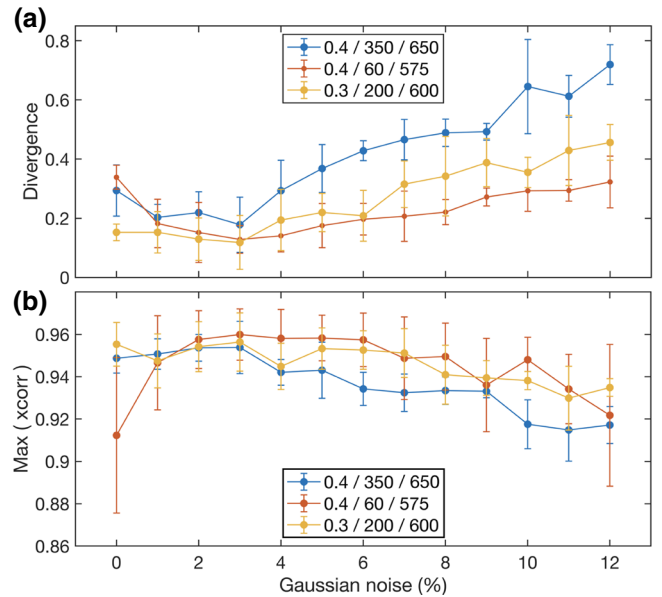


FIG. 5. (a) The divergence and (b) the maximum spectral cross-correlation coefficient (xcorr) for different levels of input noise (p). The experimental results are an average over eight realizations for each operating point and for each level of noise.

very remarkable. However, in addition, we also find a significant improvement to similar performance levels for the parameter conditions around the optimal operating point (blue and red lines). Therefore, training the RC system with input noise results in a more-robust long-term reproduction of the dynamics in an extended parameter range. This is of high practical relevance. We find that this robust reproduction of the chaotic attractor is not found when the RC system is trained with use of ridge-regression regularization. This is studied in detail in Appendix E by numerical simulations.

We argue that the improvement in the long-term reconstruction of the dynamics using an initial training with input noise is achieved because we train the RC also for the environment of the attractor manifold, resulting in higher robustness for autonomous signal generation.

The constructive role of noise can also be shown visually via attractor reconstruction. In Fig. 6, we present side by side the embedded reconstructed attractor resulting from the original time series and the ones obtained when we train the RC with the original input and the input altered with a 3% amplitude of Gaussian noise. It becomes apparent by comparing these attractors that the RC system that uses the output weights computed from the altered input yields a more accurate reconstruction of the RC chaotic attractor, retaining most of the structures present in the original one.

In addition to the prediction of the x variable of the Rössler system and the corresponding attractor reconstruction, we explore the possibility of performing *autonomous*

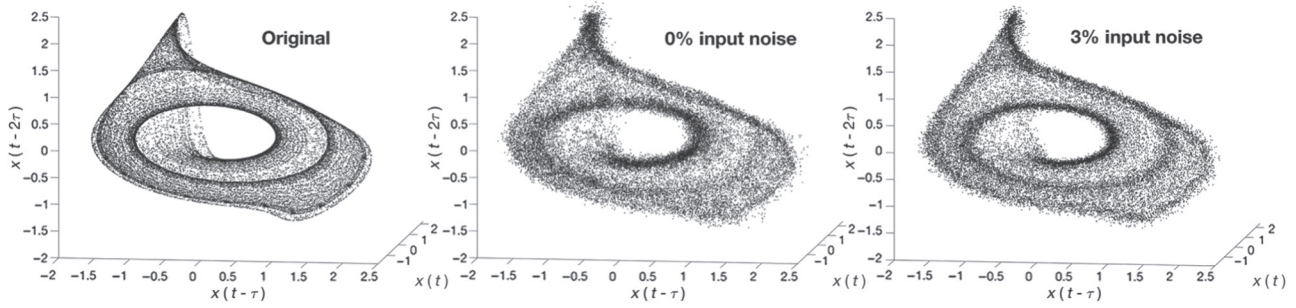


FIG. 6. Phase-space reconstructions of the Rössler chaotic attractor for the operating point $(\beta, \gamma, \Phi) = (0.4, 60, 575)$ in the space spanned by $[x(t), x(t - \tau), x(t - 2\tau)]$ with $\tau = 1.5$. The different panels show the phase-space reconstructions for the original time series and for the autonomously generated ones with different values of input noise. The phase-space reconstruction obtained for 0% and 3% input noise is obtained by our plotting of the eight different repetitions of the experiment together, which corresponds to approximately 30 000 points. We use the same number of points for the phase-space reconstruction of the original Rössler attractor.

cross-prediction of the variables y and z from the knowledge of x . During the open-loop training we obtain three sets of output weights, each of them for one of the variables of the system. As input to the reservoir we still choose the x variable of the Rössler system. In autonomous operation, we obtain three predicted values, $x(n + 1)$, $y(n + 1)$, and $z(n + 1)$, which are used for the attractor reconstruction. We find that the NRMSE for the one-step-ahead cross-prediction of the y (z) variable is 0.025 (0.04). With the autonomously generated variables x, y , and z , we verify the high quality of the experimentally reconstructed attractor, as shown in Fig. 7.

IV. FINAL REMARKS

We demonstrate that training a reservoir computer with input altered with additive Gaussian noise can be beneficial when reconstructing the long-term chaotic dynamics in a climatelike sense. Moreover, the parameter range in which high-quality chaotic attractor climate replication is found can be significantly extended. Our results highlight that the reservoir with output feedback can be understood as a high-dimensional dynamical system the stability properties of which need to be tailored. By adding noise to the input in the initial training for the short-term prediction, we include the environment of the invariant manifold of the chaotic attractor during training. Thus, the autonomous generation is less vulnerable when the predicted trajectory diverges transversely from the attractor manifold and robust replication of chaotic attractor dynamics can be achieved.

The amount of noise that needs to be added to achieve the best performance may change depending on the task. In particular, we find that for the autonomous replication of the Rössler dynamics, the best results are achieved when the input signal is altered with approximately-3%-amplitude Gaussian noise. These results are demonstrated quantitatively by two different methods based on

recurrence plots and power spectra. Supporting numerical simulations further validate the overall positive influence of the pretraining with noise for the replication of the chaotic attractor (see Appendix E). For the optimal conditions of our setup and for our target signal, we also perform autonomous cross-prediction of the variables y and z . Applying this method and using a hardware-based reservoir, we are able to cross-predict the unobserved variables of the Rössler attractor. We expect our

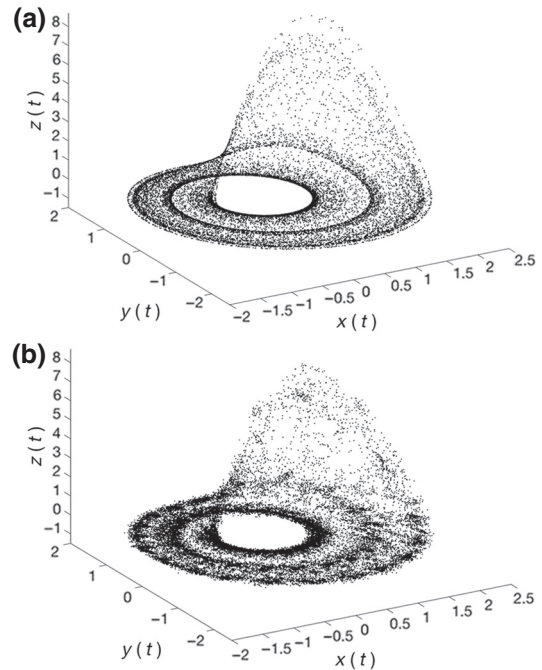


FIG. 7. Autonomous cross-prediction of all three variables of the Rössler system using only one input variable, the time series of the x variable. (a) The original Rössler attractor. (b) The reconstructed attractor obtained via autonomous cross-prediction. Each plot contains data from four repetitions of the experiment, which corresponds to approximately 15 000 points.

results to be transferable to other physical implementations of neuro-inspired information-processing schemes. Still, more experimental studies will be required to verify this point. The high-quality replication of chaotic attractor dynamics paves the way for the use of hardware-based RC systems with output feedback for arbitrary waveform generation or control loops.

ACKNOWLEDGMENTS

We thank S. Ortín and R. Vicente for valuable scientific discussions. This work was supported by the Spanish Ministry of Science, Innovation and Universities, the National Agency for Research Funding (AEI) and the European Regional Development Fund under Project No. TEC2016-80063-C3 IDEA (AEI/FEDER, UE) and the María de Maeztu program for units of excellence (Grant No. MDM-2017-0711). M.C.S. was supported by the Spanish Ministerio de Economía, Industria y Competitividad through a Ramon y Cajal Fellowship (Grant No. RYC-2015-18140).

APPENDIX A: EXPERIMENTAL SETUP

Our implementation of the delay-based scheme can be conceptually divided into several distinct blocks, as illustrated in Fig. 3. We use a hybrid system with an analog part (nonlinearity) and a digital part. The analog part represents a Mackey-Glass circuit made of electronic transistors, integrated circuits, capacitors, and resistors, as depicted in Fig. 8. This electronic circuit produces a transfer function that can be fitted to the following Mackey-Glass nonlinearity:

$$X_{\text{out}} = F(X_{\text{in}}) = \frac{CX_{\text{in}}}{1 + b^p X_{\text{in}}^p}, \quad (\text{A1})$$

where C , b , and p are parameters adjusted via the `curve_fit` function in PYTHON to approximate the experimental transfer function. The experimentally measured Mackey-Glass nonlinearity is shown together with its corresponding fit on the right-hand side in Fig. 3. Here the fit parameters correspond to $C = 2.1345$, $b = 0.0019$, and $p = 9.8212$ in Eq. (A1).

In this experimental setup, the delay-based RC implementation is working in the map limit regime (i.e., the

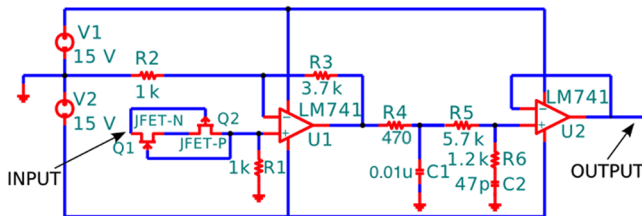


FIG. 8. The Mackey-Glass electronic implementation.

time response of the system considered $T \sim 4.7 \mu\text{s}$ is much smaller than the time separation between the virtual nodes $\theta \sim 90 \mu\text{s}$. Under these conditions, the reservoir state $r(t)$ is affected only by the input signal $x(t)$ and the state of the reservoir one delay time earlier $r(t - \tau)$. Here the coupling between virtual nodes is introduced by a mismatch between the delay time (τ) and the input sampling period (T_{in}), $\tau \neq T_{\text{in}}$ [16]. In particular, we use a ring topology such that $\tau = T_{\text{in}} + \theta$. The communication between the digital and analog parts of the experimental implementation is done via general-purpose I/O pins of the Raspberry Pi, in addition to commercial analog-to-digital-converter and digital-to-analog-converter modules with ten-bit resolution.

APPENDIX B: SIZE OF THE RESERVOIR

A key parameter for the performance of reservoir computing is the size of the reservoir. The number of nodes required depends on the task and the amount of available training data, and a larger number of nodes does not necessarily imply a better prediction. In Fig. 9, we show the computed $E = \sqrt{([y(n) - o(n)]^2)}$ for training and testing stages in the one-step-ahead task for different numbers of nodes. The training is performed with 3600 samples of the x variable of the Rössler system. Considering the error on the one-step-ahead prediction during the testing stage, we can observe that the best performance is obtained for 700 nodes. Experiments are initially performed at the operating point $(\beta, \gamma, \Phi) = (0.4, 60, 575)$ considering previous results [14].

APPENDIX C: OPTIMAL OPERATING POINT

The operating point within the Mackey-Glass nonlinearity can be tuned by varying the set of parameters (β, γ, Φ) . To identify the best parameter range of the reservoir's nonlinearity for this task, we consider the common assumption

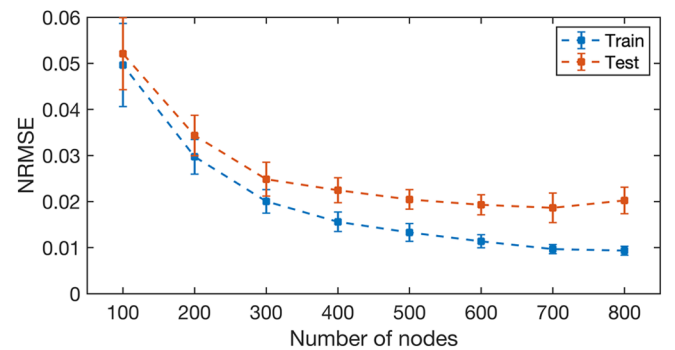


FIG. 9. Experimental results for the NRMSE during training and testing stages for the one-step-ahead prediction task as a function of the number of nodes in the reservoir. The mean and standard deviation are computed for ten realizations of the experiment.

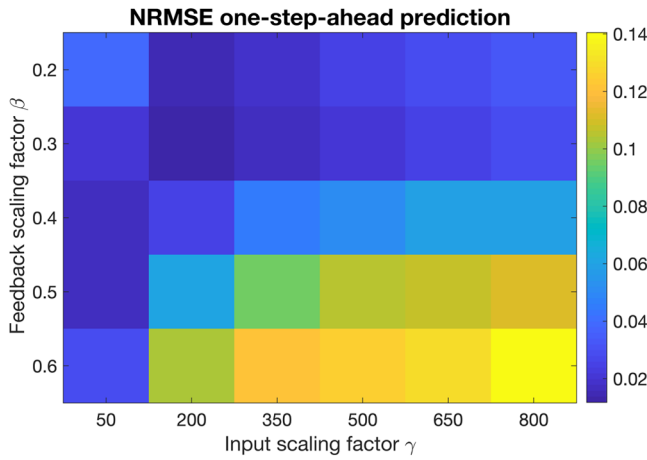


FIG. 10. Experimental results for the NRMSE during one-step-ahead prediction as a function of the feedback strength β and the input scaling γ . This scan considers the value of the phase Φ with the lowest NRMSE. From our input time series of 4000 samples, 80% of the reservoir responses are used for training and the remaining 20% are used for the one-step-ahead prediction task. The results shown here are cross-validated by our splitting the reservoir response into five data sets.

that the lower the value of the error during the short-term prediction, the better the performance of the autonomous operation. In Fig. 10, we show the NRMSE for one-step-ahead prediction in the testing stage as a function of the feedback strength β and the input scaling γ . Here we also simultaneously scan over the variable Φ , considering the value that leads to the lowest NRMSE for each combination of γ and β . We find that the optimal operating point is $(\beta, \gamma, \Phi) = (0.3, 200, 600)$.

In addition to this optimal operating point, in the main text, we consider two neighboring points with different NRMSEs during the one-step-ahead prediction: $(0.4, 60, 575)$ and $(0.4, 350, 650)$.

APPENDIX D: RECURRENCE PLOTS

A recurrence plot is a two-dimensional visualization of a sequential data set [21,22]. Considering our time series as a set of M scalar measurements

$$x = (x_1, x_2, x_3, \dots, x_M), \quad (\text{D1})$$

we can construct an m_e -dimensional vector a_k for each value in x using the Takens embedding theorem. This theorem offers a method to reconstruct a phase space very similar to that of the full solution through a delay-coordinate embedding generated using only the values of x . The vector is defined as

$$\mathbf{a}_k = (x_k, x_{k+\tau_e}, x_{k+2\tau_e}, \dots, x_{k+(m_e-1)\tau_e}), \quad (\text{D2})$$

where τ_e is the so-called embedding delay time and m_e is the embedding dimension.

A recurrence plot is obtained by considering whether a given pair of these coordinates is nearby in the embedding space. Such recurrence of the trajectory at times i and j is then pictured within a two-dimensional squared matrix R with dots, where both axes are time axes. The elements of this matrix are defined as

$$R_{ij}^{d,\varepsilon} = \Theta(\varepsilon - \|\mathbf{a}_i - \mathbf{a}_j\|), \quad (\text{D3})$$

where a_{ij} ($i, j = 1, 2, \dots, M$) are vectors of dimension m_e , ε is a threshold distance, $\|\cdot\|$ is a norm, and Θ is the Heaviside function. If the distance between a pair of coordinates in the series is less than the threshold parameter ε , then $R_{ij} = 1$, which can be plotted as a black dot. Otherwise $R_{ij} = 0$, which can be plotted as a white dot.

The challenge with recurrence plots is the optimal choice of the embedding dimension m_e [23], the embedding delay time τ_e , and the threshold distance ε [24–26]. In our study, we apply the autocovariance function to the input time series x to find the optimal value of the delay, $\tau_e = 5$. Next we chose the threshold to be a few percent of the maximum phase space (i.e., $\varepsilon = 0.45$). Finally, we chose the embedding dimension m_e to be 3. For this choice of parameters, we observe linear structures with small diagonal lines of differing length characteristic of chaotic systems, as illustrated in Fig. 11.

To quantify the different structures appearing in the recurrence plots, RQA was developed by Zbilut and Webber [27,28] and was extended with new measures of complexity by Marwan *et al.* [29]. These measures, usually computed in windows along the main diagonal, aim to study time dependencies within the time series. They are useful to detect transitions, time delays, unstable periodic orbits, and similarities between processes. For our study, we considered the following quantities:

(a) Recurrence rate. This quantifier measures the density of recurrence points in a recurrence plot (i.e., the probability that a specific state will recur).

$$R = \frac{1}{N^2} \sum_{ij=1}^N R(i, j). \quad (\text{D4})$$

(b) Determinism. This quantifier is the percentage of recurrence points that form diagonal lines in the recurrence plot of minimal length ℓ_{\min} ($\ell_{\min} \geq 2$). It is related to the predictability of the dynamical system. For a white-noise time series, its recurrence plot will be full of only single dots and very few diagonal lines. Instead, a deterministic process has a recurrence plot with very few single dots but many long diagonal lines.

$$T = \frac{\sum_{\ell=\ell_{\min}}^N \ell P(\ell)}{\sum_{\ell=1}^N \ell P(\ell)}, \quad (\text{D5})$$

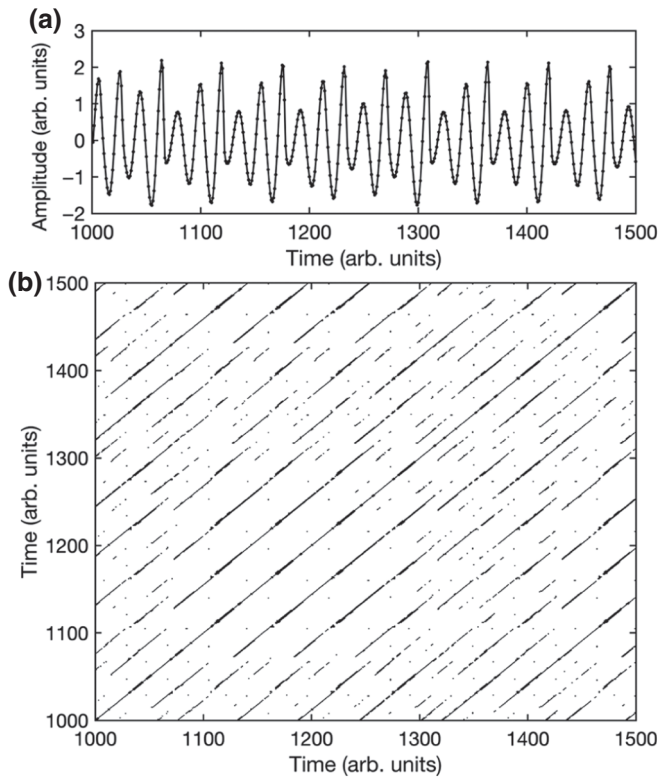


FIG. 11. (a) An interval of the time series of the x variable of the Rössler chaotic system. (b) The corresponding recurrence plot.

where $P(\ell)$ is the frequency distribution of the lengths ℓ of the diagonal lines.

(c) Averaged diagonal-line length. This measures the average length of the diagonal lines.

$$L = \frac{\sum_{\ell=\ell_{\min}}^N \ell P(\ell)}{\sum_{\ell=\ell_{\min}}^N P(\ell)}. \quad (\text{D6})$$

(d) Shannon entropy. The probability $p(\ell)$ that a diagonal line has exactly a length ℓ can be estimated from the frequency distribution $P(\ell)$ with $p(\ell) = P(\ell)/\sum_{\ell=\ell_{\min}}^N P(\ell)$. The Shannon entropy of this probability,

$$H = - \sum_{\ell=\ell_{\min}}^N p(\ell) \ln p(\ell), \quad (\text{D7})$$

is a manifestation of the complexity of the deterministic structure in the system.

We propose RQA as a method to quantify the similarity between the reconstructed and the original chaotic dynamics in the climatolike sense. In particular, we compute the recurrence rate, determinism, averaged diagonal-line length, and Shannon entropy from the recurrence plot of

TABLE II. Recurrence rate, determinism, averaged diagonal-line length, and Shannon entropy for the x variable of the Rössler chaotic system measured in the recurrence plot in Fig. 11.

R	T	L	H
0.0162	0.9356	8.8566	1.8980

the original time series, the values of which can be seen in Table II, and we compare them with the ones obtained with the autonomous replication of the reservoir computer. This quantitative measure is called “divergence,” and is defined as the sum of the absolute weighted-values differences between the different RQA measures, as shown in Eq. (3).

APPENDIX E: NUMERICAL SIMULATIONS

By performing accompanying numerical simulations, we aim to test whether our main findings are not specific to the experimental system. We can also isolate the influence of the training with additive noise at the input as opposed to using regularization in the training. Here we use ridge

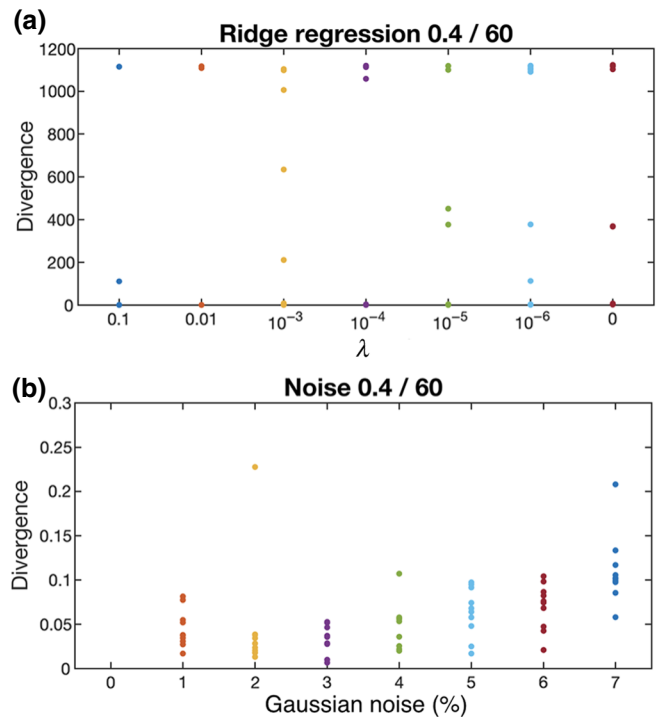


FIG. 12. Divergence of the attractor reconstruction for the operating point $(\beta, \gamma, \Phi) = (0.4, 60, 575)$. (a) The output weights are calculated with use of ridge regression for different values of λ . (b) The output weights are calculated with use of linear regression and input noise.

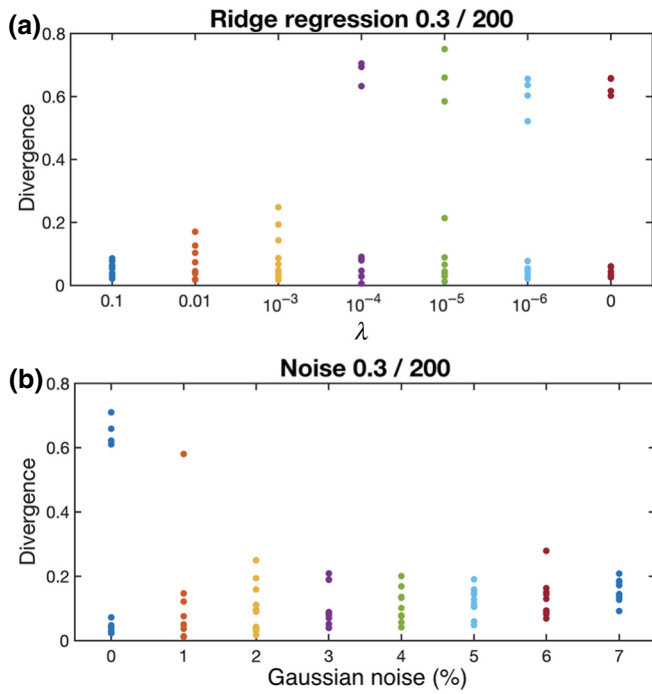


FIG. 13. Divergence of the attractor reconstruction for the operating point $(\beta, \gamma, \Phi) = (0.3, 200, 600)$. (a) The output weights are calculated with use of ridge regression for different values of λ . (b) The output weights are calculated with use of linear regression and input noise.

regression for different values of λ in Eq. (E1):

$$\sum_{i=1}^m (y_i - o_i)^2 + \lambda \sum_{j=1}^N (W_j^{\text{out}})^2. \quad (\text{E1})$$

The first term corresponds to the usual linear regression; the second one imposes the regularization.

We also extend the number of samples (length of the time series) considered during the training to avoid artifacts, if any, due to the initial conditions. In the numerical simulations, we use $m = 35\,000$ points to train the system (i.e., calculate the output weights) and 5000 points to test it. Then we close the loop, and we let the system run autonomously for another 20 000 time steps, which will eventually generate the desired chaotic attractor. We perform this procedure for ten different sets of input weights; the corresponding results are shown in Figs. 12–14. In these figures each dot corresponds to a different set of input weights.

With the results depicted in Figs. 12(a) and 12(b), we illustrate that ridge-regression regularization is not analogous to training the system by adding noise to the input time series. We find that training the system using a regularization method, independently of the value of the ridge-regression parameter λ , does not guarantee the proper reconstruction of the attractor for the operating point $\beta = 0.4$, $\gamma = 60$, $\Phi = 575$. As shown by the large

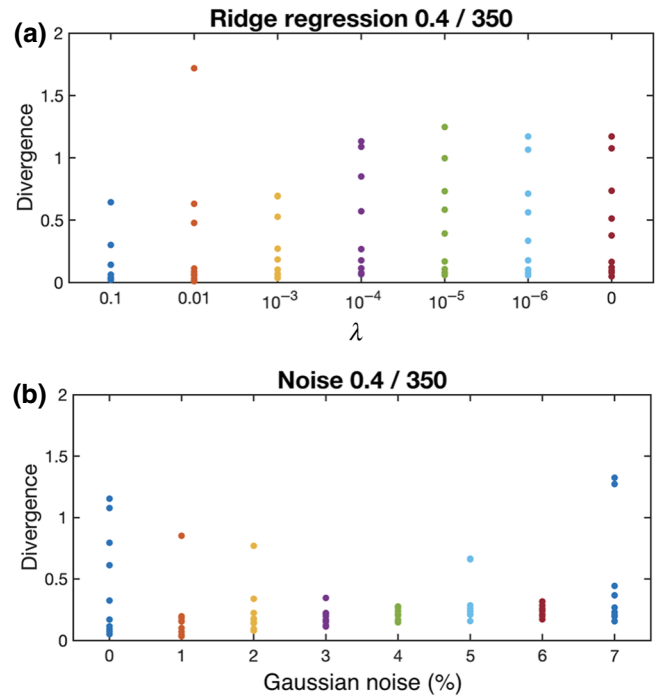


FIG. 14. Divergence of the attractor reconstruction for the operating point $(\beta, \gamma, \Phi) = (0.4, 350, 650)$. (a) The output weights are calculated with use of ridge regression for different values of λ . (b) The output weights are calculated with use of linear regression and input noise.

values of the divergence in Fig. 12(a), the autonomously generated time series no longer reproduce the dynamics of the Rössler system for certain realizations of the input weights. In such cases, we find that the reservoir approaches a fixed point, a limit cycle, or other erratic temporal behavior, represented by a cloud of points in the attractor reconstruction representation.

For the other two operating points ($\beta = 0.3$, $\gamma = 200$, $\Phi = 600$ and $\beta = 0.4$, $\gamma = 350$, $\Phi = 650$), Figs. 13 and 14 illustrate that we obtain different results when using either linear regression with input noise or ridge-regression regularization. We show that for these operating points ridge regression works for high values of λ (i.e., $\lambda \geq 10^{-3}$). The attractor reconstruction for output weights calculated with linear regression and input noise reaches a minimum around the 3%–4% of additive Gaussian noise. Our numerical results support the finding that training the reservoir with additive noise at the input yields a stable autonomous reproduction of the chaotic attractor for intermediate values of the noise strength. In turn, the robust reproduction of the chaotic attractor is significantly less likely when ridge regression is used for all the operating points.

[1] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso,

- and I. Fischer, Information processing using a single dynamical node as complex system, *Nat. Commun.* **2**, 468 (2011).
- [2] J. Torrejon, M. Riou, F. A. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, and J. Grollier, Neuromorphic computing with nanoscale spintronic oscillators, *Nature* **547**, 428 (2017).
- [3] G. Dion, S. Mejaouri, and J. Sylvestre, Reservoir computing with a single delay-coupled non-linear mechanical oscillator, *J. Appl. Phys.* **124**, 152132 (2018).
- [4] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, Parallel photonic information processing at gigabyte per second data rates using transient states, *Nat. Commun.* **4**, 1364 (2013).
- [5] L. Gammaitoni, P. Hänggi, P. Jung, and F. Marchesoni, Stochastic resonance, *Rev. Mod. Phys.* **70**, 223 (1998).
- [6] F. Sagues, Jose M. Sancho, and Jordi Garcia-Ojalvo, Coherence resonance, *Rev. Mod. Phys.* **79**, 829 (2007).
- [7] E. Mompoto, M. Ruiz-Garcia, M. Carretero, H. T. Grahn, Y. Zhang, and L. L. Bonilla, Stochastic and Coherence Resonance, *Phys. Rev. Lett.* **121**, 086805 (2018).
- [8] P. Antonik, M. Haelterman, and S. Massar, Brain-Inspired Photonic Signal Processor for Generating Periodic Patterns and Emulating Chaotic Systems, *Phys. Rev. Appl.* **7**, 054014 (2017).
- [9] D. Canaday, A. Griffith, and D. Gauthier, Rapid time series prediction with a hardware-based reservoir computer, *Chaos* **28**, 123119 (2018).
- [10] Z. Lu, B. R. Hunt, and E. Ott, Attractor reconstruction by machine learning, *Chaos* **28**, 061104 (2018).
- [11] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach, *Phys. Rev. Lett.* **120**, 024102 (2018).
- [12] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, *Science* **304**, 78 (2004).
- [13] J. D. Farmer and J. J. Sidorowich, Predicting Chaotic Time Series, *Phys. Rev. Lett.* **59**, 845 (1987).
- [14] M. C. Soriano, S. Ortín, L. Keuninckx, L. Appeltant, J. Danckaert, L. Pesquera, and G. Van der Sande, Delay-based reservoir computing: Noise effects in a combined analog and digital implementation, *IEEE Trans. Neural Netw. Learn. Syst.* **26**, 388 (2015).
- [15] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, L. Pesquera, C. R. Mirasso, and I. Fischer, Photonic information processing beyond turing: An optoelectronic implementation of reservoir computing, *Opt. Express* **20**, 3241 (2012).
- [16] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, Optoelectronic reservoir computing, *Sci. Rep.* **2**, 287 (2012).
- [17] R. Martinenghi, S. Rybalko, M. Jacquot, Y. K. Chembo, and L. Larger, Photonic Non-linear Transient Computing with Multiple-delay Wave- Length Dynamics, *Phys. Rev. Lett.* **108**, 244101 (2012).
- [18] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, All-optical reservoir computing, *Opt. Express* **20**, 22783 (2012).
- [19] S. Ortín, M. C. Soriano, L. Pesquera, D. Brunner, D. San-Martín, I. Fischer, C. R. Mirasso, and J. Gutiérrez, A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron, *Sci. Rep.* **5**, 14945 (2015).
- [20] N. Marwan, M. C. Romano, M. Thiel, and J. Kurths, Recurrence plots for the analysis of complex systems, *Phys. Rep.* **438**, 237 (2007).
- [21] T. K. March, S. C. Chapman, and R. O. Dendy, Recurrence plot statistics and the effect of embedding, *Phys. D: Nonlinear Phenomena* **200**, 171 (2005).
- [22] J. P. Zbilut Jr., and L. C. Webber, in *Wiley Encyclopedia of Biomedical Engineering* (Wiley-InterScience, Hoboken, N.J., 2006).
- [23] F. M. Atay and Y. Altıntaş, Recovering smooth dynamics from time series with the aid of recurrence plots, *Phys. Rev. E* **59**, 6593 (1999).
- [24] G. M. Mindlin and R. Gilmor, Topological analysis and synthesis of chaotic time series, *Phys. D* **58**, 229 (1992).
- [25] M. Koebe and G. Mayer-Kress, Use of recurrence plots in the analysis of time-series data, *Proc. SFI Stud. Sci. Complexity* **XXI**, 361 (1992).
- [26] J. P. Zbilut and C. L. Webber, Jr., Embeddings and delays as derived from quantification of recurrence plots, *Phys. Lett. A* **171**, 199 (1992).
- [27] J. P. Zbilut, Embeddings and delays as derived from quantification, *Phys. Lett. A* **171**, 199 (1992).
- [28] C. L. Webber, Jr. and J. P. Zbilut, Dynamical assessment of physiological systems and states using recurrence plot strategies, *J. Appl. Physiol. Respir. Environ. Exerc. Physiol.* **76**, 965 (1994).
- [29] N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan, and J. Kurths, Recurrence-plot-based measures of complexity and their application to heart rate-variability data, *Phys. Rev. E* **66**, 026702 (2002).