Using reservoir computers to distinguish chaotic signals

T. L. Carroll*
US Naval Research Laboratory, Washington, DC 20375, USA

(Received 27 August 2018; published 8 November 2018)

Several recent papers have shown that reservoir computers are useful for analyzing and predicting dynamical systems. Reservoir computers have also been shown to be useful for various classification problems. In this work, a reservoir computer is used to identify one out of the 19 different Sprott systems. An advantage of reservoir computers for this problem is that no embedding is necessary. Some guidance on choosing the reservoir computer parameters is given. The dependance on number of points, number of reservoir nodes, and noise in identifying the Sprott systems is explored.

DOI: 10.1103/PhysRevE.98.052209

I. INTRODUCTION

Describing chaotic signals is difficult because of their complex nature. If an experiment produces a chaotic signal, some way to describe the signal is necessary to detect changes in the experiment. There have been a number of methods published for comparing or identifying chaotic signals [1–8], but most of these methods require embedding the signal in a phase space, which requires knowledge of the embedding dimension and delay. Phase space embeddings are also sensitive to noise, as individual points are displaced by noise in multiple dimensions, so interpoint distances are not accurate. There are other methods for characterizing attractors, such as fractal dimension, Lyapunov exponents, and linking numbers [9]. These methods are commonly used because in theory they are invariant under orientation preserving diffeomorphisms, so that a change in the embedded variable or the embedding method should not change the measurement. In practice, there are well known problems when applying these standard methods to real data; see, for example, [10].

There has been quite a bit of recent work on using reservoir computers to model and predict chaotic systems [11–15] so it is known that reservoir computers are useful for analyzing chaotic signals. A reservoir computer is simply a high-dimensional dynamical system that is driven by a signal to be analyzed. Usually the dynamical system is created by connecting a set of nonlinear nodes in a network so that the entire dynamical system has a stable fixed point. The dynamical system then responds to the input signal of interest, acting as a nonlinear filter. Training of the reservoir computer comes about by forming a linear combination of many signals from the dynamical system to fit a training signal; for example, in [14], the dynamical system is driven by the Lorenz x signal, and a set of signals from the dynamical system is fitted to the Lorenz z signal. The fit coefficients are saved. In computational mode, the same dynamical system is then driven by a Lorenz x signal with different initial conditions. Using the previously fitted coefficients to make a linear combination of signals from the dynamical system, the z signal corresponding to the particular x signal is reproduced.

The reservoir computer does introduce additional complexity in that the dynamical system typically contains from 100 to 1000 nodes. Long term research focuses on implementing reservoir computers as analog systems [15,16], creating an advantage in terms of computational speed. Another complication in applying reservoir computers is that there is no complete theory of how reservoir computers operate, so choosing parameters for a reservoir computer to solve a particular problem proceeds by trial and error. There has been some work towards a theory of reservoir computing [11,17–19], with some emphasis on the concept of generalized synchronization.

In this paper, I begin by describing a reservoir computer and how to train the computer. I then choose particular parameters for the reservoir computer using signals from the Sprott B chaotic system [20]. Next I describe how to create a coefficient vector that is characteristic of a particular Sprott system, and I show how to use these coefficient vectors to determine from which Sprott system a particular signal originated, and I characterize the error performance of this signal identification.

II. RESERVOIR COMPUTING

Reservoir computing is a branch of machine learning [16,21,22]. A reservoir computer consists of a set of nonlinear nodes connected in a network. The set of nodes is driven by an input signal, and the response of each node is recorded as a time series. A linear combination of the node response signals is then used to fit a training signal. Unlike other types of neural networks, the network connecting the nonlinear nodes does not vary; only the coefficients used to fit the training signal vary.

The reservoir computer used in this work is described by

$$\frac{d\mathbf{R}}{dt} = \lambda[\alpha\mathbf{R} + \beta\mathbf{R}^2 + \gamma\mathbf{R}^3 + \mathbf{A}\mathbf{R} + \mathbf{W}s(t)]. \quad (1)$$

$\mathbf{R}$ is vector of node variables, $\mathbf{A}$ is a sparse matrix indicating how the nodes are connected to each other, and $\mathbf{W}$ is a vector

*Thomas.Carroll@nrl.navy.mil

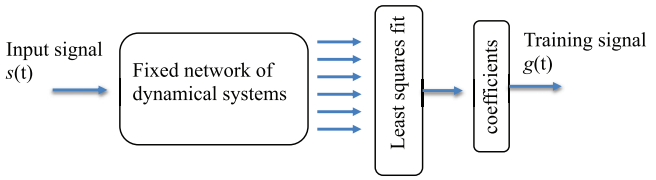

FIG. 1. Block diagram of a reservoir computer. The input signal $s(t)$ drives a fixed network of dynamical nodes. The time varying signal from the nodes is fitted to the training signal $g(t)$ by a least squares fit.

that describes how the input signal $s(t)$ is coupled to each node. The constant λ is a time constant, and there are M nodes. For all the simulations described here, α, β, and γ are set to make the network stable; that is, the network has a stable fixed point with a large basin of attraction.

The particular reservoir computer used here is arbitrary, and other types of nodes can also be used. The main requirements for a reservoir computer are that the nodes are nonlinear and that the network of nodes has a stable fixed point, so that in the absence of an input signal the network does not oscillate [22].

Figure 1 is a block diagram of a reservoir computer.

To train the reservoir computer, an input signal $s(t)$ and a training signal $g(t)$ were chosen and Eq. (1) was numerically integrated. The first part of the response of the reservoir computer was discarded as a transient, and the next N time series points $r_i(t)$, $i = 1, \ldots, M$, from each node were combined in an $N \times (M+1)$ matrix,

$$\Xi = \begin{bmatrix} r_1(1) & r_1(2) & \ldots & r_1(N) \\ \vdots & & & \\ r_M(1) & r_M(2) & \ldots & r_M(N) \\ 1 & 1 & \ldots & 1 \end{bmatrix}. \tag{2}$$

The last row of Ξ was set to 1 to account for any constant offset in the fit. The training signal is fitted by

$$g(t) = \sum_{j=1}^{M} c_j r_j(t) \tag{3}$$

or

$$g(t) = \Xi \mathbf{C}, \tag{4}$$

where $g(t) = [g(1), g(2), \ldots, g(N)]$ is the training signal.

The matrix Ξ is decomposed by a singular value decomposition

$$\Xi = \mathbf{U}\mathbf{S}\mathbf{V}^T, \tag{5}$$

where $\mathbf{U}$ is $N \times (M+1)$, $\mathbf{S}$ is $N \times (M+1)$ with nonnegative real numbers on the diagonal and zeros elsewhere, and $\mathbf{V}$ is $(M+1) \times (M+1)$.

The pseudoinverse of Ξ is constructed as

$$\Xi_{\text{inv}} = \mathbf{V}\mathbf{S}^{'}\mathbf{U}, \tag{6}$$

where $\mathbf{S}^{'}$ is an $(M+1) \times (M+1)$ diagonal matrix, where the diagonal element $S^{'}_{i,i} = S_{i,i}/(S^2_{i,i} + \delta^2)$, where $\delta = 1 \times 10^{-5}$ is a small number used for ridge regression to prevent overfitting.

The fit coefficient vector is then found by

$$\mathbf{C} = \Xi_{\text{inv}} g(t). \tag{7}$$

The coefficient vector $\mathbf{C}$ will be used as a feature vector to identify individual signals. The difference between signals i and j is computed as

$$\Delta_{ij} = \sum_{k=1}^{M+1} \sqrt{\mathbf{C}_i^2(k) - \mathbf{C}_j^2(k)}, \tag{8}$$

where ${\mathbf{C}_i}^2(k)$ is the kth component of the coefficient vector for signal i.

The training error E_T may be computed from

$$E_T = \frac{\|\Xi\mathbf{C} - g(t)\|}{\|g(t)\|}. \tag{9}$$

The training error is used as a measure of how well the training signal $g(t)$ may be reconstructed from the input signal $s(t) = [s(1), s(2), \ldots, s(N)]$.

Classification

The coefficient vector $\mathbf{C}$ can be used to classify signals. For each class of signal, a driving signal and a training signal are chosen; for example, one may drive the reservoir with a Lorenz x signal and train on a Lorenz z signal. The combination of the fixed reservoir network and the coefficient vector $\mathbf{C}(\text{Lorenz}_1)$ found by training on the z signal form a classifier for the Lorenz system. One may then form a classifier for the Rossler system by driving the same network with a Rossler x signal and training on a Rossler z signal. Note that the reservoir network never changes. A new coefficient vector $\mathbf{C}(\text{Rossler}_2)$ is found by training on the Rossler z signal.

If a Lorenz system with the same parameters is started with different initial conditions, the same reservoir network may be driven with the new Lorenz x signal and trained on the new Lorenz z signal to yield a new coefficient vector $\mathbf{C}(\text{Lorenz}_2)$. The new Lorenz signal may be classified by computing Δ_{ij} [Eq. (8)] where i indicates $\mathbf{C}(\text{Lorenz}_2)$ and j indicates $\mathbf{C}(\text{Lorenz}_1)$ or $\mathbf{C}(\text{Rossler}_1)$. The smaller value of Δ_{ij} will be found when taking the difference between the two coefficient vectors for the Lorenz system.

III. SPROTT SYSTEMS

Sprott [20] found a family of 19 different chaotic systems defined by 3-dimensional ODEs with 1 or 2 quadratic nonlinearities. Sprott was looking for simple chaotic systems for both theoretical and practical reasons, as different chaotic systems were needed for testing potential applications of chaos. Sprott considered three-dimensional ordinary differential equations with quadratic nonlinearities. He limited the nonzero coefficients to a total of 6 to make the search feasible, and searched this 6-dimensional space for solutions with a positive Lyapunov exponent. This search yielded 19 distinct chaotic systems, which he labeled A through S. This group of chaotic systems is a useful test set for our signal comparison methods.

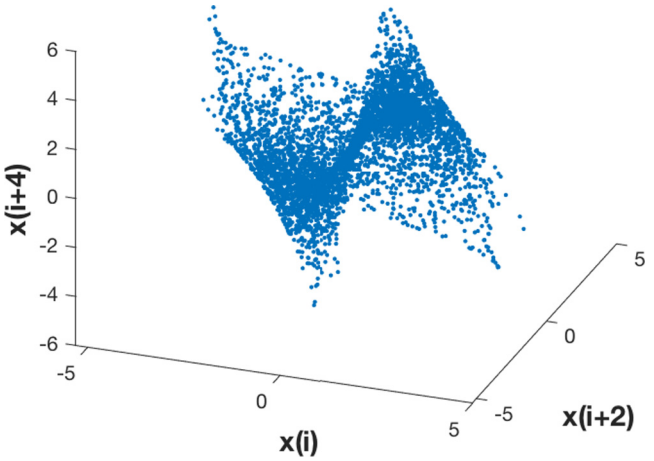

FIG. 2. Embedded time series signal for the Sprott B attractor with an embedding delay of 2.

Each set of ODEs for the Sprott systems was integrated using a fourth order Runge-Kutta integrator with a time step of 0.01. The integrator output was decimated by keeping every 50th point to produce a time series.

As an example, the Sprott B system was described by the differential equations

$$\frac{dx}{dt} = yz, \quad \frac{dy}{dt} = x - y, \quad \frac{dz}{dt} = 1 - xy. \tag{10}$$

.

Figure 2 is a plot of the embedded attractor for the Sprott B system.

Figure 3 is the autocorrelation of the $x(t)$ signal from the Sprott B system. The autocorrelation will be used in setting the parameters for the reservoir computer.

IV. RESERVOIR COMPUTER PARAMETERS

There is currently no theory for designing a reservoir computer to solve a particular problem, so parameter choice for Eq. (1) must proceed by trial and error. The training error E_T from Eq. (9) was used as a metric to judge the accuracy of the reservoir computer: the smaller E_T, the better the computer. The parameters that produce the smallest E_T may not be the best parameters for calculating the difference Δ_{ij} between two signals, but optimizing for Δ_{ij} requires that we

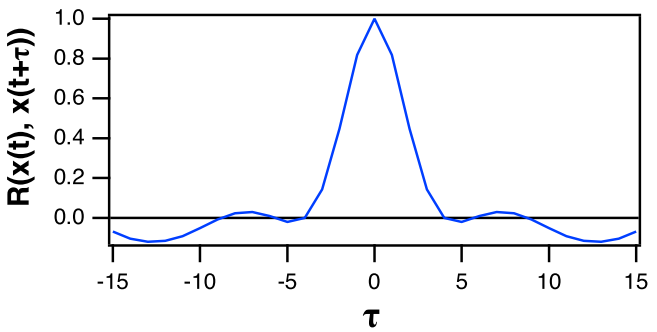

FIG. 3. Autocorrelation $R(x(t), x(t+\tau))$ for the $x(t)$ signal from the Sprott B system.

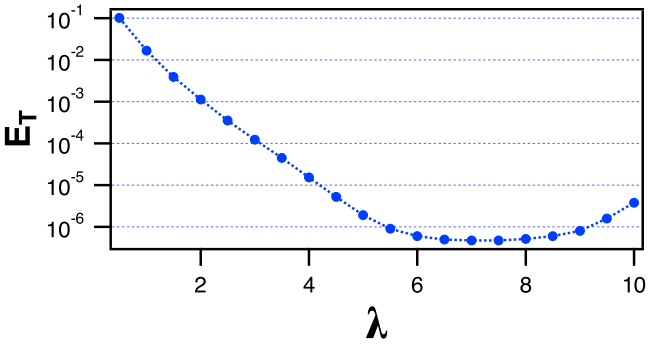

FIG. 4. Mean of the training error E_T for all of the 19 different Sprott systems as the time constant λ defined in Eq. (1) is varied. The input signal $s(t) = x(t)$ for all 19 Sprott systems, and the training signal $g(t)$ was also equal to $x(t)$.

know in advance that the two signals are different. We may not know in advance if the signals are the same or different.

First, the parameters $\alpha = -3$, $\beta = 1$, and $\gamma = -1$ in Eq. (1) were chosen so that the network was stable. The number of nodes was set at $M = 100$. Next, the specific network matrix **A** and input coupling vector **W** were determined. The parameters **A** and **W** were determined by choosing 100 randomly selected **A** and **W** pairs and keeping the pair that yielded the lowest training error T_E.

For the determination of **A** and **W**, the input signal $s(t)$ and the training signal $g(t)$ were set to the $x(t)$ variable from the Sprott B system. Both $s(t)$ and $g(t)$ were normalized to have a mean of 0 and a standard deviation of 1. The time constant λ was set to an arbitrary value of 1.

The input signal $s(t)$ was 6000 points long, and after driving the reservoir, the first 1000 points from all signals were discarded as a transient. One hundred random realizations of **A** were generated from a uniform random distribution between ± 1. The matrix **A** was sparse, with 20% of its elements nonzero, and all nodes had at least one connection to other nodes. **A** was normalized so that the largest absolute value of the real part of its eigenvalues was 0.5. Another 100 random realizations of **W** were generated from a uniform random distribution between ± 0.5.

The training error [Eq. (9)] was recorded for each random network configuration and the **A** and **W** pair that gave the lowest training error E_T was retained as part of the optimum parameter set.

Next the value for the time constant λ, which determined the frequency response of the reservoir, was set. The reservoir computer responds to a finite band of frequencies, so to make sure this band of frequencies was optimal for analyzing the Sprott system signals, the time constant was varied between 0.5 and 10 and the training error was computed with the input and training signals $s(t)$ and $g(t)$ equal to the $x(t)$ signals for each of the 19 Sprott systems.

Figure 9 is a plot of the mean of the training error E_T for all 19 different Sprott systems as the time constant λ is varied.

Figure 4 shows that the mean of the training error for all the Sprott systems is small for $\lambda = 7$, so the time constant for the reservoir computer in Eq. (1) is set to 7. The minima for each of the individual Sprott systems occurred at roughly

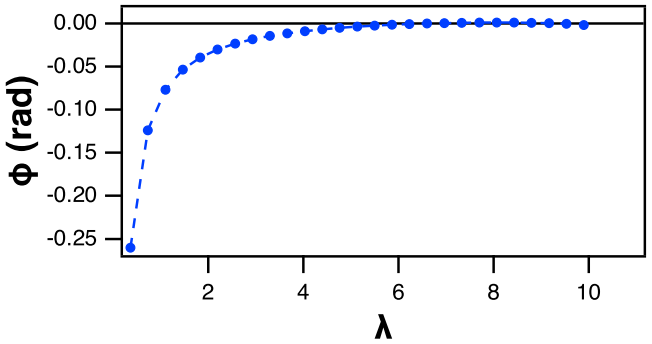

FIG. 5. Phase shift ϕ for a single node being driven by a sine wave, as a function of the node time constant λ. ϕ has its closest approach to 0 for $\lambda = 6.6$.

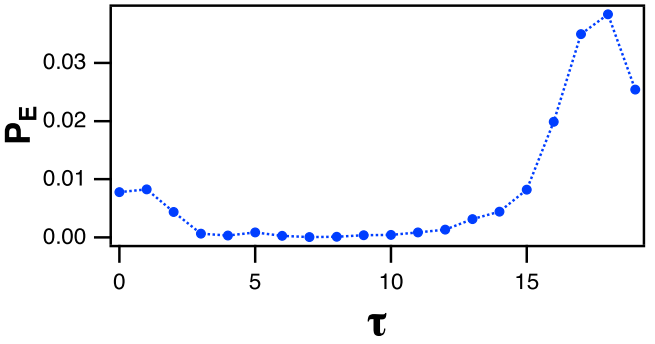

FIG. 6. Probability of making an error P_E in identifying the 19 Sprott systems when the input signal to the reservoir computer $s(t) = x(t)$ and the training signal is $g(t) = x(t + \tau)$.

the same value of λ, so the minimum of the mean was a good approximation. If the value of λ at which the minimum training error occurred was very different for different Sprott systems, then the signals could probably be distinguished by their frequency content alone, and no reservoir computer would be necessary.

Figure 5 shows a possible reason that the minimum training error occurs at $\lambda = 7$. For Fig. 5, a single node was driven by a sine wave with a period of 20 points, which is the approximate number of points per cycle for the Sprott systems in this paper. Figure 5 shows the shift in phase between the node variable $r(t)$ and the driving sine wave $s(t)$. The closest approach to zero for the phase shift ϕ between the driving sine wave and the single node response occurred at $\lambda = 6.6$, near the value of $\lambda = 7$ that minimized the training error.

V. SIGNAL IDENTIFICATION

It has been shown that reservoir computers are useful for signal identification or classification, for speech signals (for example, [23]), or image recognition [24]. For signal fitting, the smallest training error will undoubtedly come when the training signal $g(t)$ is equal to the input signal $s(t)$; such is not the case for the error in identifying signals.

To identify the Sprott signals, the reservoir computer of Eq. (1) was driven with the signal $x(t)$ from each of the Sprott systems, while the training signal $g(t)$ was set equal to $x(t + \tau)$. The reservoir computer of Eq. (1) was numerically integrated with a fourth order Runge-Kutta integration routine with a time step of 0.1. The first 1000 time steps were discarded and the next 5000 time steps from each node were used to find the fitting coefficients $\mathbf{C}$ as in Eq. (7).

For each Sprott system, a time series consisting of 600 000 points of the $x(t)$ signal was generated and divided into 100 sections of 6000 points each. The reservoir computer node variables $\mathbf{R}(t)$ were all initialized to an initial value of 0. For each section, the reservoir computer was driven by the input signal $s(t) = x(t)$ and the first 1000 points of the network response variables $r_i(t)$, $i = 1, \ldots, M$, were dropped to eliminate the transient. The fitting coefficients for each Sprott system for each of the 100 sections were found according to Eqs. (2)–(7).

For each of the 19 Sprott systems there were therefore 100 sets of coefficients $\mathbf{C}_i(k)$, where $i = \mathrm{A}, \mathrm{B}, \ldots, \mathrm{S}$ indicates the particular Sprott system and $k = 1, 2, \ldots, 100$ indicates the section of the Sprott signal.

The difference between two Sprott systems was defined in Eq. (8) as Δ_{ij}. The difference between two sections from two Sprott systems is

$$\Delta_{ij}(l_1, l_2) = \sum_{k=1}^{M+1} \sqrt{\mathbf{C}_i^2(l_1, k) - \mathbf{C}_j^2(l_2, k)}, \qquad (11)$$

where $l_1 = 1, 2, \ldots, 100$ and $l_2 = 1, 2, \ldots, 100$ indicate the different sections of the Sprott signals, i and j indicate the different Sprott systems, and $k = 1, 2, \ldots, M + 1$ indicates the particular component of the coefficient vector.

When comparing the Sprott systems, when sections l_1 and l_2 are compared, if the minimum value of $\Delta_{ij}(l_1, l_2)$ is not $\Delta_{ii}(l_1, l_2)$,

$$\min[\Delta_{ij}(l_1, l_2)] < \Delta_{ii}(l_1, l_2), \quad j \neq i, \quad l_1 \neq l_2, \qquad (12)$$

then an error is recorded. The comparisons are made for all the coefficient vectors of all the sections of all 19 Sprott systems, and the probability P_E of making an error in correctly identifying each Sprott system was recorded.

Figure 6 is a plot of the probability P_E of making an error in identifying the Sprott systems as the delay τ in the training signal $g(t) = x(t + \tau)$ varies. Figure 6 shows that the smallest error in identifying the Sprott systems occurs when $\tau > 0$, so the training signal $g(t)$ does not match the input signal $x(t)$. The minimum error occurs for values of τ between 4 and 7.

The identification error probability is lower when the training signal $g(t)$ is delayed from the input signal $x(t)$ because the relation between the delayed signal and the nondelayed signal contains information unique to the particular chaotic system. Knowing what comes later in time for a particular signal gives more information than just knowing the signal at a particular time. To quantify this extra knowledge, the mutual information between the input signal $x(t)$ and the training signal $g(t)$ was computed.

To compute the mutual information, each signal was transformed into a symbolic time series using the ordinal pattern method [25]. Each signal was divided into windows of 4 points, and the points within the window were sorted to establish their order; for example, if the points within a

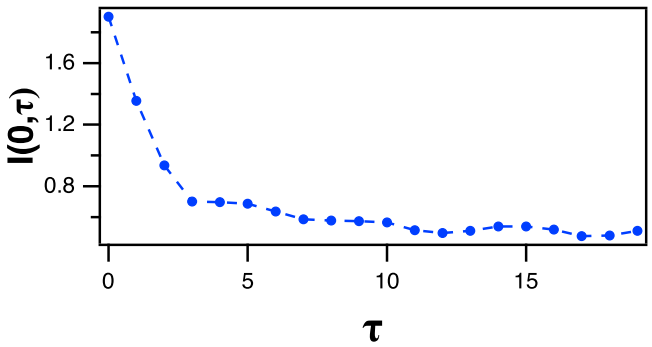

FIG. 7. Mean of the mutual information $I(0,\tau)$ between $x(t)$ and $x(t+\tau)$ for all 19 Sprott systems.

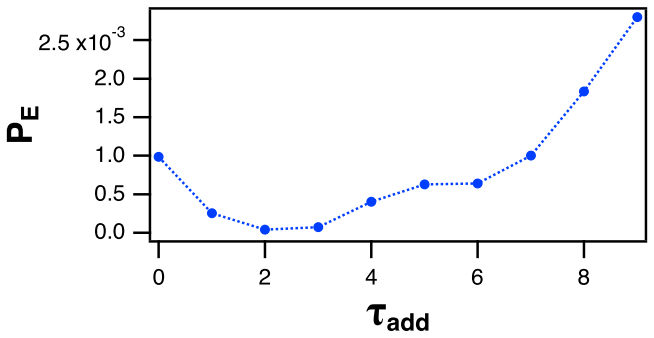

FIG. 8. Probability of identification error P_E as a function of the added delay τ_{add} for the 19 Sprott systems. The input signal was $s_i(t) = x_i(t)$, while the training signal was $g_i(t) = x_i(t+\tau_i+\tau_{\text{add}})$, where the index i indicated the particular Sprott system. The delay τ_i was the delay for which the autocorrelation of the $x(t)$ signal from Sprott system i first dropped below 0 or reached its first minimum.

window were 0.1, 0.3, −0.1, 0.2, the ordering would be 2, 4, 1, 3. Each possible ordering of points in $x(t)$ represented a symbol $\sigma_q(0), q = 1,\ldots,N_{s0}$, where N_{s0} was the number of possible symbols. Each symbol in the delayed signal $x(t+\tau)$ was $\sigma_q(\tau), q = 1,\ldots,N_{s\tau}$. The probabilities $p(\sigma_q(0))$ and $p(\sigma_q(\tau))$ were found for each symbol. The mutual information between the signal $x(t)$ and the delayed version $x(t+\tau)$ was

$$I(0,\tau) = \sum_{q1=1}^{N_{s0}} \sum_{q2=1}^{N_{s\tau}} p[\sigma_{q1}(0),\sigma_{q2}(\tau)] \log_{10} \times \left(\frac{p[\sigma_{q1}(0),\sigma_{q1}(\tau)]}{p[\sigma_{q1}(0)]p[\sigma_{q2}(\tau)]} \right). \quad (13)$$

Figure 7 is a plot of the mean of the mutual information between the input signal $x(t)$ and the delayed version $x(t+\tau)$ for all 19 Sprott systems. The mutual information between $x(t)$ and $x(t+\tau)$ decreases sharply for $\tau \leqslant 3$ and then starts to level off. The delayed signal $x(t+\tau)$ has new information not present in $x(t)$, although the amount of new information does not increase as rapidly for $\tau > 3$.

The probability of identification error plot in Fig. 6 increases for $\tau > 7$. The Sprott systems produce chaotic signals, so for long delays, $x(t+\tau)$ will be uncorrelated with $x(t)$. As an example, Fig. 3 is a plot of the autocorrelation $R(x(t),x(t+\tau))$ for the $x(t)$ signal from the Sprott B system. The autocorrelation for the Sprott B system pictured in Fig. 3 first drops below 0 for $\tau = 4$. For the other Sprott systems, the autocorrelation for the $x(t)$ signal drops below 0 for delays ranging from $\tau = 2$ to $\tau = 6$, except for the Sprott C system, where the autocorrelation does not drop below 0 until $\tau = 33$, but the autocorrelation for the Sprott C system does have its first minimum at $\tau = 5$. If the delay τ for the training signal $g(t) = x(t+\tau)$ is increased by too much, the training signal becomes uncorrelated with the input signal and the probability of identification error will increase.

To find the ideal delay for the training signal $g(t) = x(t+\tau)$ for each of the 19 Sprott systems, τ_i, $i = \text{A, B, }\ldots\text{, S}$, was set equal to the delay for which the autocorrelation for that signal first dropped below 0 or reached its first minimum. An additional delay τ_{add} was then added to each τ_i, so each of the 19 training signals was $g_i(t) = x_i(t+\tau_i+\tau_{\text{add}})$. Figure 8 shows the probability of identification error P_E as a function of the added delay for the Sprott systems. The identification error is minimized for $\tau_{\text{add}} = 2$. It is evident that the optimum delay for the training signal $g(t) = x(t+\tau)$ is slightly greater than the delay for which the first minimum occurs in the autocorrelation function. Choosing this delay maximizes the new information provided by the training signal but keeps the training signal from becoming too uncorrelated with the input signal. This simple rule is similar to the conventional wisdom for choosing the delay window in a delay embedding, that the window length should be equal to the delay at which the first zero (or first minimum) in the autocorrelation is seen.

To summarize the preceding section: Longer delays mean that the delayed training signal contains more new information (Fig. 7), but this new information is counterbalanced by the decreasing correlation between input and training signals (Fig. 3). This is the same trade-off that comes into play when choosing the delays for delay embedding of a signal.

A. Number of data points

Figure 9 shows the error in identifying the 19 Sprott systems P_E as a function of the total number of points used N. The total number of points includes the 1000-point transient. As a comparison, probability of error from the density method of [6,7] is also plotted. The reservoir computer method required fewer points to identify the Sprott systems, and it did not require that the signal be embedded in a phase space.

The error performance of the reservoir computer of Eq. (1) was also compared to the error performance of a reservoir computer with a different type of node. The most commonly used node type in the reservoir computer literature is a sigmoid nonlinearity, so a reservoir computer described by Eq. (14) was also simulated:

$$\mathbf{R}_i(n+1) = \alpha\left[\frac{1}{1+e^{-\mathbf{R}_i(n)}} + \mathbf{A}\mathbf{R}(n) + \mathbf{W}s(t)\right]. \quad (14)$$

The parameter $\alpha = 0.35$. The network connection matrix $\mathbf{A}$ and the input coupling vector $\mathbf{C}$ were the same as in Eq. (1), as were the input and training signals. The probability of error for identifying the 19 Sprott systems using the reservoir computer with a sigmoid nonlinearity is also plotted in Fig. 9.

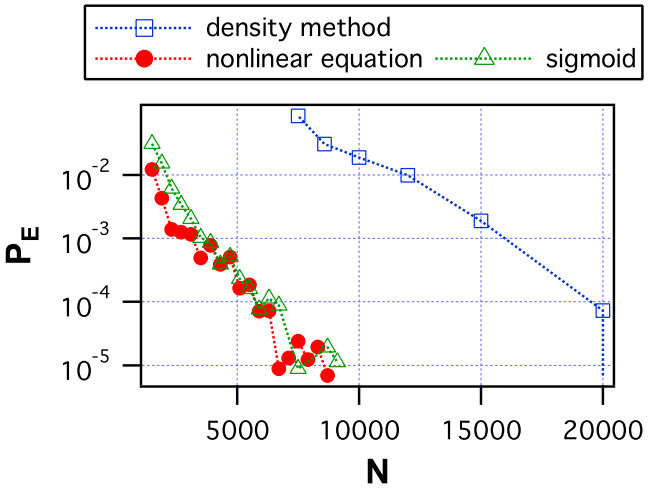

FIG. 9. Probability of error P_E as a function of number of points N in the input time series $s(t) = x(t)$ for identifying the Sprott systems using a reservoir computer as described in Eq. (1), labeled as "nonlinear equation." The input signal was $s(t) = x(t)$ for each of the Sprott systems, while the training signal $g(t) = x(t + \tau)$, where τ was the delay for the first 0 or first minimum of the autocorrelation function for each Sprott system plus 2 time steps. The figure also shows the probability of error for identifying the Sprott systems from the density method of [6,7], labeled as the "density method" and from a reservoir computer using a sigmoid node described in Eq. (14) ("sigmoid"). The number of points N for the reservoir computers includes the 1000 point transient.

The probability of error when the sigmoid nonlinearity is used is approximately the same as when the nonlinear equation (1) is used.

B. Number of nodes

It seems that the number of nodes M in the reservoir computer should make a difference in the probability of error in identifying the Sprott systems, and Fig. 10 confirms this suspicion. Each time the number of nodes M was changed, optimal values for the connection matrix **A** and the input vector **W** were determined as in Sec. IV. The identification

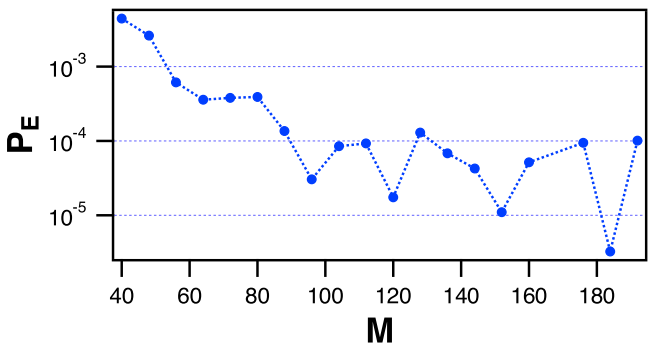

FIG. 10. Probability of error P_E in identifying the 19 Sprott systems using the reservoir computer of Eq. (1) when the number of nodes M in the network was varied. The delay τ in the training signal $g(t) = x(t + \tau)$ was set to $\tau = \tau_i + 2$, where τ_i the delay for which the autocorrelation for Sprott system i first drops below 0 or has its first minimum.

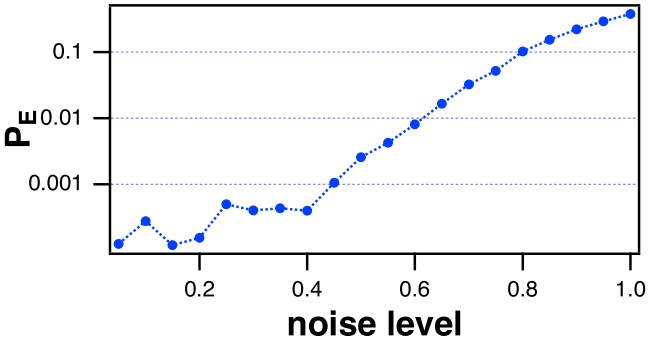

FIG. 11. Probability of error in identifying the 19 Sprott systems with added Gaussian white noise. The noise level is the ratio of the noise standard deviation to the signal standard deviation. The delay τ in the training signal $g(t) = x(t + \tau)$ was set to $\tau = \tau_i + 2$, where τ_i is the delay for which the autocorrelation for Sprott system i first drops below 0 or has its first minimum.

error P_E as a function of the number of reservoir computer nodes M is plotted in Fig. 10.

Figure 10 shows that the scatter in the probability of identification P_E is very large as the number of nodes increases. Most likely this scatter is caused by the fact that the network connection matrix **A** is not optimal. As the number of nodes M increases, the number of elements in **A** increases as M^2, so the probability of randomly generating an optimum matrix **A** from a fixed number of random realizations decreases, meaning that **A** is less likely to be optimum as M increases.

C. Added noise

Figure 11 shows the probability of identification error P_E for identifying the 19 Sprott systems as noise is added to the input signal; $s(t) = x(t) + \eta(t)$, where $\eta(t)$ is Gaussian white noise. The reservoir computer was described by Eq. (1). The noise level in Fig. 11 is the ratio of the standard deviation of the noise signal to the standard deviation of $x(t)$. For each of the Sprott systems the delay τ used to determine the training signal $g(t) = s(t + \tau)$ was the delay for which the autocorrelation for that Sprott system first dropped below 0 (or had its first minimum) plus 2 time steps. The noise level on the training signal $g(t)$ is the same as the noise level on the input signal $s(t)$.

Figure 11 shows that the reservoir computer of Eq. (1) is robust to moderate amounts of added noise.

VI. SUMMARY

Reservoir computers are useful for identifying chaotic signals, as the example in this paper shows. Using a reservoir computer, it was possible to correctly identify signals from the 19 different Sprott systems with a error probability lower than that in a method that used density to identify chaotic systems [6,7]. An advantage of the reservoir computer method is that no embedding is required, so it is not necessary to estimate dimension or delay.

One drawback to using reservoir computers is that there is no theory to guide the selection of reservoir computer parameters. The strategy used in this paper was to choose input and training signals $x(t)$ and $g(t)$ and vary the reservoir

parameters to minimize the training error T_E, Eq. (9). This approach gives reasonable parameters, but it is not optimum, since the parameters where the training error is minimized may not be the parameters that minimize the error in identifying different systems, P_E.

The reservoir computer method does require numerically integrating a network of M nonlinear systems, and as Fig. 10 shows, larger values of M give lower error probabilities. Implementing a reservoir computer on a digital computer is slow, although the different nodes may be integrated in parallel. The real promise of reservoir computing is that the nodes may be implemented with analog systems, in which case speed increases over digital computing are possible.

[1] J. D. Farmer and J. J. Sidorowich, Phys. Rev. Lett. **59**, 845 (1987).

[2] G. Sugihara and R. M. May, Nature (London) **344**, 734 (1990).

[3] M. Casdagli, Physica D **35**, 335 (1989).

[4] B. R. Hunt, E. J. Kostelich, and I. Szunyogh, Physica D **230**, 112 (2007).

[5] R. Brown, N. F. Rulkov, and E. R. Tracy, Phys. Rev. E **49**, 3784 (1994).

[6] T. L. Carroll and J. M. Byers, Phys. Rev. E **93**, 042206 (2016).

[7] T. L. Carroll and J. M. Byers, in *Proceedings of the 4th International Conference on Applications in Nonlinear Dynamics* (Springer, Cham, Switzerland, 2017), Vol. 6, pp. 139–149.

[8] R. Gilmore, Rev. Mod. Phys. **70**, 1455 (1998).

[9] H. D. I. Abarbanel, R. Brown, J. J. Sidorowich, and L. S. Tsimring, Rev. Mod. Phys. **65**, 1331 (1993).

[10] E. Bradley and H. Kantz, Chaos **25**, 097610 (2015).

[11] Z. Lu, B. R. Hunt, and E. Ott, Chaos **28**, 061104 (2018).

[12] R. S. Zimmermann and U. Parlitz, Chaos **28**, 043118 (2018).

[13] P. Antonik, M. Gulina, J. Pauwels, and S. Massar, Phys. Rev. E **98**, 012215 (2018).

[14] Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, Chaos **27**, 041102 (2017).

[15] G. Van der Sande, D. Brunner, and M. C. Soriano, Nanophotonics **6**, 561 (2017).

[16] M. Lukoševičius and H. Jaeger, Comput. Sci. Rev. **3**, 127 (2009).

[17] M. C. Soriano, Physics **10**, 12 (2017).

[18] S. Marzen, Phys. Rev. E **96**, 032308 (2017).

[19] M. Inubushi and K. Yoshimura, Sci. Rep. **7**, 10199 (2017).

[20] J. C. Sprott, Phys. Rev. E **50**, R647 (1994).

[21] H. Jaeger and H. Haas, Science **304**, 78 (2004).

[22] G. Manjunath and H. Jaeger, Neural Comput. **25**, 671 (2013).

[23] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer, Opt. Express **20**, 3241 (2012).

[24] A. Jalalvand, K. Demuynck, W. D. Neve, and J.-P. Martens, Neurocomputing **277**, 237 (2018).

[25] C. Bandt and B. Pompe, Phys. Rev. Lett. **88**, 174102 (2002).