

On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD F

Cite as: Chaos **31**, 013108 (2021); <https://doi.org/10.1063/5.0024890>

Submitted: 14 August 2020 • Accepted: 30 November 2020 • Published Online: 04 January 2021

 Erik Bollt

COLLECTIONS

 This paper was selected as Featured



[View Online](#)



[Export Citation](#)



[CrossMark](#)

ARTICLES YOU MAY BE INTERESTED IN

[Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **27**, 121102 (2017); <https://doi.org/10.1063/1.5010300>

[Attractor reconstruction by machine learning](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 061104 (2018); <https://doi.org/10.1063/1.5039508>

[Reservoir observers: Model-free inference of unmeasured variables in chaotic systems](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **27**, 041102 (2017); <https://doi.org/10.1063/1.4979665>



Chaos
Special Topic: Nonlinear Model Reduction From Equations and Data
Submit Today!

On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD

Cite as: Chaos 31, 013108 (2021); doi: 10.1063/5.0024890

Submitted: 14 August 2020 · Accepted: 30 November 2020 ·

Published Online: 4 January 2021



View Online



Export Citation



CrossMark

Erik Bolta^{a)} 

AFFILIATIONS

Department of Electrical and Computer Engineering, Clarkson University, Potsdam, New York 13699, USA and Clarkson Center for Complex Systems Science (C³S²), Potsdam, New York 13699, USA

^{a)}Author to whom correspondence should be addressed: boltetm@clarkson.edu

ABSTRACT

Machine learning has become a widely popular and successful paradigm, especially in data-driven science and engineering. A major application problem is data-driven forecasting of future states from a complex dynamical system. Artificial neural networks have evolved as a clear leader among many machine learning approaches, and recurrent neural networks are considered to be particularly well suited for forecasting dynamical systems. In this setting, the echo-state networks or reservoir computers (RCs) have emerged for their simplicity and computational complexity advantages. Instead of a fully trained network, an RC trains only readout weights by a simple, efficient least squares method. What is perhaps quite surprising is that nonetheless, an RC succeeds in making high quality forecasts, competitively with more intensively trained methods, even if not the leader. There remains an unanswered question as to why and how an RC works at all despite randomly selected weights. To this end, this work analyzes a further simplified RC, where the internal activation function is an identity function. Our simplification is not presented for the sake of tuning or improving an RC, but rather for the sake of analysis of what we take to be the surprise being not that it does not work better, but that such random methods work at all. We explicitly connect the RC with linear activation and linear readout to well developed time-series literature on vector autoregressive (VAR) averages that includes theorems on representability through the Wold theorem, which already performs reasonably for short-term forecasts. In the case of a linear activation and now popular quadratic readout RC, we explicitly connect to a nonlinear VAR, which performs quite well. Furthermore, we associate this paradigm to the now widely popular dynamic mode decomposition; thus, these three are in a sense different faces of the same thing. We illustrate our observations in terms of popular benchmark examples including Mackey–Glass differential delay equations and the Lorenz63 system.

Published under license by AIP Publishing. <https://doi.org/10.1063/5.0024890>

The power and success of artificial neural networks (ANNs) have been profound across many disciplines, including dynamical systems. A leader among methodologies for forecasting has been the recurrent neural network (RNN) for aspects of memory. However, because of the large number of parameters to train to data observations, and likewise the nonlinear nature of the associated optimization process, the training phase can be computationally extremely intensive. The echo-state reservoir computing (RC) concept is a significant simplification where only the output weights are trained and in a manner that allows for a straightforward and cheap least squares method. The rest of the weights,

those of input layers and those of inner layers, are simply selected randomly. It is clear that this would be cheaper to train, but what is not clear and perhaps a surprise is that it would work at all, but work it does. With a simplification of the concept to allow for a linear activation function, while the performance is not quite as good but it does still work, now we are able to analyze in detail the role of the randomly selected parameters and how there is still freedom in fitting a well defined time-series forecasting model, which, in fact, is equivalent to the well developed theory of vector autoregression (VAR). Within the VAR and related vector moving average (VMA) theory, we recall the Wold

theorem that allows us to discuss the representation, and now, as we show, it is relevant to the RC for machine learning. Also, with this description, we are able to connect to the recently highly popular dynamic mode decomposition (DMD) concept. While we do observe that the fully linear version of the RC, and so the corresponding VAR, does make reasonable short-term forecasts, a linear RC with a quadratic readout significantly improves forecasts, and even apparently once errors do occur, they seem more true to the true nature of the original system. In the spirit of the linear RC plus linear readout yields a VAR, we show that a linear RC with a (Hadamard) quadratic readout yields a nonlinear VAR (NVAR) that includes monomials of all quadratic forms.

I. INTRODUCTION

Artificial neural networks (ANNs) have emerged as a core and powerful technology in machine learning,^{25,52,62,63,65} which is well suited for the supervised learning in data-driven science and engineering, specifically including for forecasting problems in complex dynamical systems.^{15,28,40,42,48,50,53} However, the most straightforward feedforward ANN with backpropagation for training concepts can be extremely expensive to optimize the data, even considering important recent innovations, such as stochastic gradient descent or hardware breakthroughs, such as GPU-based processing. Recurrent neural network (RNN) concepts are particularly suitable for temporal data from a dynamical system,^{5,6,22,28,47,74} as they naturally embed temporal information, especially the long short-term memory (LSTM) approach demonstrates excellent fidelity,^{20,22,41,88,89,94} but these are especially expensive to train fully.⁶⁹

The reservoir computing (RC)^{44,58,87} and the closely related echo-state network (ESN)^{43,57} and liquid state machine (LSM)^{35,60} have emerged as a special variant of an RNN, where only the output layer is trained rather than the entire network of weights. As such, this requires only a simple and efficient least squares estimation rather than the more expensive full nonlinear optimization associated with fully training an RNN. Nonetheless and perhaps a most surprising outcome is that, despite this gross simplification, the forecasting capability can still be competitive even for chaotic or spatiotemporally complex problems.^{18,19,30,56,70,89,96} Specifically, an RC thrives when a full state observation is available, while fuller and more expensive variants of RNN, especially the LSTM, would be considered highly perming, especially when only a reduced variable set is available.^{88,89} Still, the RCs are popular surely because of their simplicity to train and perhaps, in part, because of their undeniable even if surprising fidelity.

The purpose of this work is to offer at least a partial explanation as to how an RC can be such a successful and general universal dynamical system for forecasting such a wide array of systems despite randomly “trained” read-in and inner weights. In other words, we work to better understand “where does the randomness go?” The purpose of this paper is not specifically to build a new method or to improve the current method, but to explain what is perhaps surprising is that the RC method works at all. In doing so, we challenge the concept with a simplified linear activation functional version for the sake that this allows our simplified analysis throughout, and even if this version has reduced fidelity, we show

that it does still have theoretical reasons that it still works, which we are now in a position to describe in detail. Our simplification to a linear activation function also allows us to explicitly write the RC as a VAR, serving as a bridge to the more theoretically well established time-series theory and also to DMD. Nonetheless, we do describe a simple scenario where the linear RC is expected to perform no worse than the widely used nonlinear counterpart in terms of efficiency, but the linear version we can now show in detail without ambiguity when there will be a good forecasting version by tying it to the representation theory of VAR as seen through the Wold theorem. While the linear activation reservoir and the linear readout is explicitly connected to a VAR and results at least for short-term forecasts are reasonable, we also show that linear reservoir with quadratic readout (as quadratic readout has become popular⁷⁰) is equivalent to a NVAR and this turns to perform quite well.

There have been few explanations as to how despite the random construction, an RC works so well, but notably.^{17,24,32} Usually alternatively, we find in the literature a collection of descriptions as to how to choose random networks as the inner layers regarding sparsity^{34,81} or regarding the design of the spectral radius for linear stability^{19,34,45} and the echo property.²⁹ An especially strong result comes from the study of Volterra series, by Boyd and Chua,¹⁴ where it was proven that a finite-dimensional linear dynamical system with a nonlinear readout, even a polynomial readout, can approximate very general signals. A new analysis by Hart *et al.* shows that regularized echo state machines make good universal approximators of ergodic dynamical systems³⁹ and furthermore give generalized embedding results³⁸ reminiscent of the classical Taken’s embedding theorem.⁸³ Also, recently, it has been shown that fading memory leads to universality in Gonon and Ortega.³³

In the spirit of still incomplete theoretical basis as to the underlying success of RC, we allow a simplified version of RC with linear activation functions for which we are able to more fully identify the inner workings of how the RC can be a universal forecasting machine even for time-series from complex and chaotic dynamical systems. We show that by this simplification, the RC still works, albeit with reduced performance quality, but nonetheless, the purpose here being theoretical explanation of how such a simple system of only training the readout is possible. We offer this variant as a theoretical construction. By this interpretation, we will also be able to connect the RC concept to other theoretically more matured theories. Specifically, the theory of autoregression (AR) from time-series analysis and moving averages (MA), together called ARMA,^{13,21,68,72,80,84} are found in the Wold theorem,⁹² which we show is directly related to the RC concept. The vector formulation of these,^{59,73} called vector autoregression (VAR) and vector moving averages (VMAs), is also connected by a corresponding Wold theorem. Furthermore, we describe a relationship to the recently highly popular dynamic mode decomposition (DMD),^{8,49,53,78,91} which is an empirical formulation of the Koopman spectral theory.^{3,8,11} Therefore, while we do not offer this simplified RC for performance over other approaches, we hope that this work will serve to shed light on how the simplified RC approach is capable of providing useful time-series forecasts and likewise as a suggestion as to how the general RC is successful. There are related concepts concerning how an RNN is closely related to a NARMA model of a stochastic process (nonlinear autoregressive moving average), which

are found in Ref. 23. While this paper is mostly motivated to describe connections between different approaches, the machine learning RC approaches, the econometrics time-series VAR approach, and also the dynamical system operator theoretical DMD approach, we show reasonable but not excellent forecasting ability of the linear RC with linear readout equivalent of a VAR. However, we do go on to connect a linear RC with quadratic readout (as quadratic readout popular for reasons of matching signal parity so it is described⁷⁰), which we show explicitly can be written as a quadratic NVAR. That nonlinearities of the reservoir may be usefully moved to the output layer has been pointed out as a possibility and of practical use when building a photonic device implementation in Ref. 86.

This paper is organized as follows. In Sec. II, we describe the nature of the data as derived from a stochastic process. In Sec. III, we review the standard RC concept, and we demonstrate it already with time-series data from a Mackey–Glass differential delay equation. Section IV is the heart of this paper, where we first present that a linear activation function allows the RC to be stated as a linear recursion, and, therefore, fitting just the readout weights can proceed in a manner such that despite random read-in and inner weights, there is a well-posed problem. Then, in this form, we are able to directly relate the linear RC solution to the classical VAR(k) solution. As such, we are then able to enlist statistical time-series forecasting theory for forecasting stochastic processes so that the Wold theorem that guarantees a VMA can then be translated to a VAR. Furthermore, the associated companion form of a VAR(1) usefully states the full vectorized problem. In Sec. VII, we note that the companion form of the VAR(1) is reminiscent of prior work for another famous concept in data-driven dynamical systems, which is the time-delay formulation of a DMD–Koopman analysis. In Sec. VIII, we present two classical examples, the Mackey–Glass differential delay equation and the Lorenz63 ordinary differential equation, that compare aspects of a full nonlinear RC and the linear variant of an RC. We will consider the issue of fading memory in Sec. IX. Finally, in Sec. VI, we show that a linear RC but with a Hadamard quadratic readout is equivalent to a quadratic NVAR of all monomial quadratic terms, analogous to the earlier result of a VAR.

II. THE DATA AS SAMPLED FROM A STOCHASTIC PROCESS

For data-driven forecasting problems, we require data from a process, including from a deterministic or otherwise from a stochastic dynamical system.¹² A process, stated,

$$\{X_t : t \in T\} \quad (1)$$

is in terms of a collection of random variables; X_t on a common probability space; and (Ω, \mathcal{B}, P) , where Ω is the sample space, \mathcal{B} the σ -algebra, and P a corresponding probability measure. T is a “time” index set and commonly it is chosen as either \mathbb{R} or \mathbb{Z} or subsets. For the sake of discussing finite samples of data, we emphasize maps, which may well be from discretely sampling a flow. A data set from such a process samples \mathbf{x}_{t_i} of X_{t_i} , stated as a time sorted sample, $\{\mathbf{x}_i\}_{i=1}^N$, $t_1 < t_2 < \dots < t_N$, using indexing notation, $\mathbf{x}_i := \mathbf{x}_{t_i}$. Uniform timing is also a simplifying assumption, $h = t_{i+1} - t_i$, for all $t_i \in T$. Assuming a vector real valued time-series, of dimension d_x ,

$\{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^{d_x}$. Data derived from a flow, say,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad (2)$$

may be collected by a stroboscopic map,

$$\mathbf{x}_{i+1} = \mathbf{F}_t(\mathbf{x}_i) = \mathbf{x}(t + \tau) = \mathbf{x}(t) + \int_t^{t+\tau} \mathbf{f}(\mathbf{x}(s))ds. \quad (3)$$

Suppressing the stroboscopic time t , this is a discrete time map \mathbf{F} ; likewise, other Poincaré maps may be useful for flight between the surface of section, and random dynamical systems may also be relevant.^{12,79} An underlying principle here is that the data should be “long enough”; likewise, a general failing of any data-driven machine learning method for forecasting a stochastic process will tend to do much better in terms of interpolation than extrapolation. Generalizing, to allow for out of sample, forecasts will tend to fare much better when the point to the forecasts is close to other observed inputs. Said another way, the quality of results can be brittle, depending as much upon curating a representative data set as the details of the method used to avoid that struggle between fitting between observations and overfitting and too far out of sample.

As a matter of presenting examples, we will highlight two classic problems that remain popular in benchmarking for machine learning in the recent literature. These will be

- the Mackey–Glass differential delay equations [Eq. (60)] and
- the Lorenz63 system [Eq. (61)],

both of which will be presented in detail in Sec. VIII. In Fig. 1, we show early in this presentation for the sake of context, a time-series data set of the Mackey–Glass system, from Eq. (60), to stand in as a typical data set. This problem is a useful benchmark, and it is often used as such^{2,9,27,36,61,66,93} perhaps because it is a well-known chaotic process but also for the sake of dimensional complexities that we recall in Sec. VIII A.

III. REVIEW OF THE TRADITIONAL RC WITH A NONLINEAR SIGMOIDAL ACTIVATION FUNCTION

In this section, we review the standard and fully nonlinear RC method, by which we mean, including the use of a nonlinear activation function $q(s)$. In this context, $q(s)$ is usually taken to be a sigmoidal function such as the hyperbolic tangent function. However, in Sec. IV, we will challenge these steps including simplifying to the identity function, $q(s) = s$.

Assuming the training data, $\{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^{d_x}$, the reservoir computing RNN is stated,

$$\begin{aligned} \mathbf{r}_{i+1} &= (1 - \alpha)\mathbf{r}_i + \alpha q(\mathbf{A}\mathbf{r}_i + \mathbf{u}_i + \mathbf{b}), \\ \mathbf{y}_{i+1} &= \mathbf{W}^{out}\mathbf{r}_{i+1}. \end{aligned} \quad (4)$$

The hidden variable $\mathbf{r}_i \in \mathbb{R}^{d_r}$ is generally taken to be of a much higher dimension $d_r > d_x$ by a linear lifting transformation,

$$\mathbf{u}_i = \mathbf{W}^{in}\mathbf{x}_i, \quad (5)$$

and \mathbf{W}^{in} is a randomly selected matrix $d_r \times d_x$ of weights (see Fig. 2). \mathbf{A} is also a linear transformation, as randomly chosen square matrix $d_r \times d_r$ of weights, which should be designed with certain properties such as spectral radius for convergence^{19,45} or sparsity^{56,70,89}

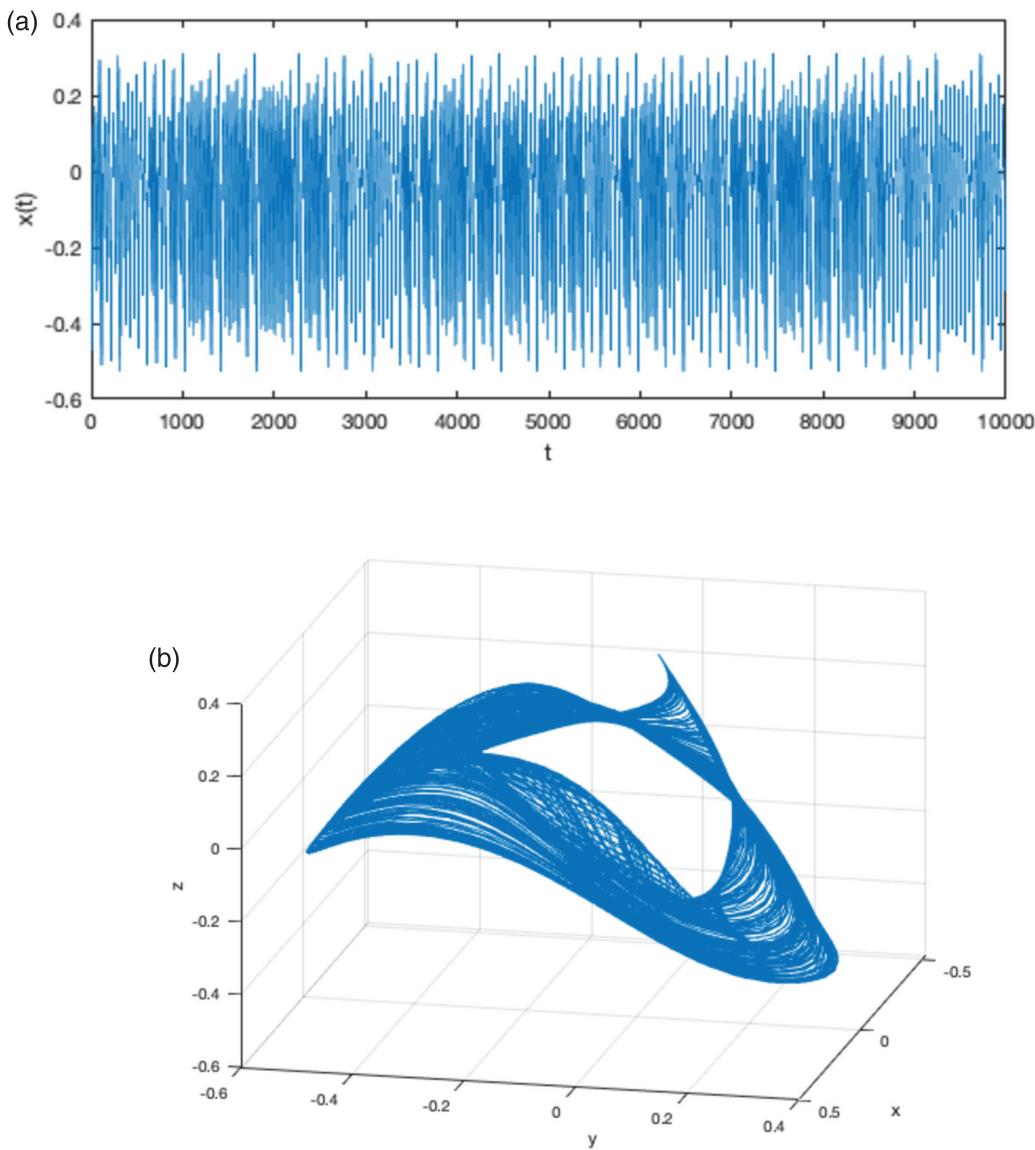


FIG. 1. Time-series acquired from the Mackey–Glass differential delay equation [Eq. (60)] has become a standard example for time-series forecasting for benchmarking data-driven methods since it is dynamically rich and high-dimensional and, therefore, challenging. (Top) Time-series, index. (Bottom) Three-dimensional projection in delay coordinates, $(x(t), x(t - \tau), x(t - 2 * \tau))$, and $\tau = 20$. A sample of $N = 10\,000$ data points is chosen as the training data set.

or otherwise consideration of the “echo-state” property.¹⁷ Likewise, the readout is by a linear transformation using a $d_x \times d_r$ matrix of weights \mathbf{W}^{out} . However, \mathbf{W}^{out} , and only \mathbf{W}^{out} , is trained to the data, allowing for forecasts y_i given data $\mathbf{x}_i \in \mathbb{R}^{d_x}$, which is the major simple aspect of RC since it can be done by a simple and cheap least squares computation. Finally, $q : \mathbb{R} \rightarrow \mathbb{R}$ is an “activation” function, using the phrasing from machine learning in the neural network community to mimic the concept of a biological network that fires when a voltage has reached a threshold. Popular choices

include $q(s) = \tanh(s)$, meaning a componentwise application of the scalar hyperbolic tangent function when s is multivariate. Other activations are popular in general neural network theory, including other sigmoidal functions, and also the ReLu function in certain contexts but not so commonly in RC.³⁰ $0 \leq \alpha \leq 1$ serves to slow down the RC to moderate stability of the fitting, but we will restrict to $\alpha = 1$ in this paper as outside the purpose of challenging the concept of explaining how the RC may work in a special case of identity q in which case, nonzero α can be considered absorbed into

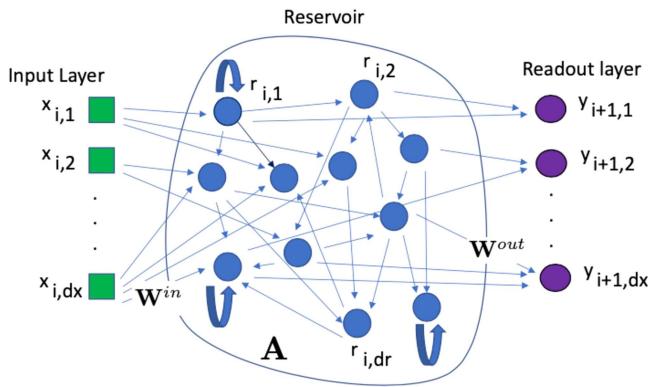


FIG. 2. Reservoir computing (RC) as defined in Eq. (4), including a randomly selected $d_r \times d_x$ read-in matrix, \mathbf{W}^{in} from $d_x \times 1$ states vector \mathbf{x} , a randomly selected $d_r \times d_r$ inner layer recurrence matrix \mathbf{A} for inner states $d_r \times 1$ vector \mathbf{r} , and the $d_x \times d_r$ trained readout matrix \mathbf{W}^{out} .

the random \mathbf{A} ; $(1 - \alpha)\mathbf{r} + \alpha\mathbf{A}\mathbf{r} = ((1 - \alpha)\mathbf{I} + \alpha\mathbf{A})\mathbf{r}$, and since \mathbf{A} is chosen randomly, then $((1 - \alpha)\mathbf{I} + \alpha\mathbf{A})$ may be an alternative random selection. Finally, \mathbf{b} serves as an offset for activation, which is useful in some context, but it is also not relevant for our needs for the same reason we choose $\alpha = 1$ and $\mathbf{b} = 0$.

What is remarkable about RC is that the usual hard work of optimally developing a full RNN is almost entirely skipped. Instead of learning \mathbf{W}^{in} and \mathbf{A} optimally fitted to the data, these seemingly very important matrices are simply picked randomly. This is enormous savings over what would usually be inherently hard to handle since the parameters are composed within the nonlinear activation q and require at least a gradient descent optimization of backpropagation in a high-dimensional and likely multi-peaked optimization space. Almost any matrix distribution may plausibly be due, but several different recipes are suggested. We say “recipe” rather than algorithm since these are descriptions of successful observations in practice rather than a product of mathematical theory that is still not complete. Here, we choose the entries of \mathbf{A} uniformly, $A_{ij} \sim U(-\beta, \beta)$, with β to scale the spectral radius, but other choices are common, notably for sparsity. The read-in matrix is also chosen uniformly randomly, $\mathbf{W}_{ij}^{in} \sim U(0, \gamma)$, with $\gamma > 0$ chosen to scale the inner variables \mathbf{r} .

The crucial aspect of the simplification that makes reservoir computing so easy and computationally efficient is that training to the output becomes just a linear process. The cheap and simple least squares solution is easily handled directly by matrix computations. Let

$$\begin{aligned} \mathbf{W}^{out} &= \arg \min_{\mathbf{V} \in \mathbb{R}^{d_x \times d_r}} \|\mathbf{X} - \mathbf{VR}\|_F \\ &= \arg \min_{\mathbf{V} \in \mathbb{R}^{d_x \times d_r}} \sum_{i=k}^N \|\mathbf{x}_i - \mathbf{V}\mathbf{r}_i\|_2, \quad k \geq 1. \end{aligned} \quad (6)$$

Notation here is standard that $\|\cdot\|_F$ denotes the Frobenius-norm of the matrix, which is the least squares equivalent of the least squares matrix parameter estimation problem. The data $\{\mathbf{x}_i\}_{i=1}^N$ are stated as

a $d_x \times N - k$ array,

$$\mathbf{X} = [\mathbf{x}_{k+1} | \mathbf{x}_{k+2} | \dots | \mathbf{x}_N] = [\mathbf{V}\mathbf{r}_{k+1} | \mathbf{V}\mathbf{r}_{k+2} | \dots | \mathbf{V}\mathbf{r}_N] = \mathbf{VR}, \quad k \geq 1 \quad (7)$$

are the forecasts to \mathbf{X} to be optimized in least squares by \mathbf{W}^{out} , processed through the RC,

$$\mathbf{R} = [\mathbf{r}_{k+1} | \mathbf{r}_{k+2} | \dots | \mathbf{r}_N], \quad k \geq 1. \quad (8)$$

While $k = 1$ is allowable, here, for theoretical development in Secs. IV–IX, we allow for larger $k \geq 1$, describing memory. In practice, a ridge regression (Tikhonov regularization with least squares regularity^{7,30,31,70}) is used to mitigate overfitting, the solution of which may be written formally,

$$\mathbf{W}^{out} := \mathbf{XR}^T(\mathbf{RR}^T + \lambda\mathbf{I})^{-1}. \quad (9)$$

Here, \cdot^T is the matrix transpose, \mathbf{I} is the identity matrix, and the choice of regularity parameter is $\lambda \geq 0$. We will write a regularized pseudo-inverse with the notation

$$\mathbf{R}_\lambda^\dagger := \mathbf{R}^T(\mathbf{RR}^T + \lambda\mathbf{I})^{-1}. \quad (10)$$

In Appendix A, we review the matrix theory as to how to form regularized pseudo-inverses such as $\mathbf{R}_\lambda^\dagger$ by a regularized singular value decomposition (SVD) in terms of regularized singular values such as $\sigma_i / (\sigma_i^2 + \lambda)$ obtained from the singular values σ_i from the SVD of \mathbf{R} .

In Fig. 3, we show an example of an RC machine from data obtained from the Mackey-Glass differential delay equations [Eq. (60)]. We see fitting for $N = 10\,000$ data points $x(t)$, $d_x = 1$, regularizing parameter $\lambda = 1.0 \times 10^{-8}$ and fitting for constant time offset. Fit and true data are shown to be so close that in fact, the blue fit curve hides the red true data curve. Also shown are several (7) of the $d_r = 500$ hidden variables $r(t)$. The fit matrix \mathbf{A} is randomly chosen with entries from a uniform distribution and then scaled so that the spectral radius $\rho(\mathbf{A}) = 1$. The random matrix \mathbf{W}_{in} is also chosen uniformly, scaled so that x values lead to \mathbf{r} in $[-0.6, 0.6]$. In Fig. 4, the trained RC is used to forecast into the future. We see small errors grow in scale, as illustrated by the bottom error curve. Results from an RC forecasting for the Lorenz63 system are presented in Sec. VIII B; notably, the forecasting quality degrades more quickly in part due to known large Lyapunov instability of that system.

What is amazing is despite the fact that RC may seemingly be a gross oversimplification of the RNN concept, it still seems to work quite well. Also from our experience, it is generally stable in that it is somewhat insensitive to the parameters and hyperparameters of the fitting process even if the level of quality does depend on these. Furthermore, once it starts to make larger errors, the kind of dynamics it produces is still plausible alternative wave forms of the process. Nonetheless, there are some parameter choices to make; notably, $d_r > d_x$ must be “large enough,” but how big is not well understood. Furthermore, the nature of the underlying distribution of matrices \mathbf{W}^{in} and \mathbf{A} is not fully understood. We hope to contribute some general perspective as to why an RC may work at all. Our goal here is not specifically to improve performance and admitting that the corresponding VAR makes reasonable short-term forecasts, but perhaps no better than that, as illustrated in our examples. However, we go on in Sec. VI, with details in Appendix B, to show that fitting a

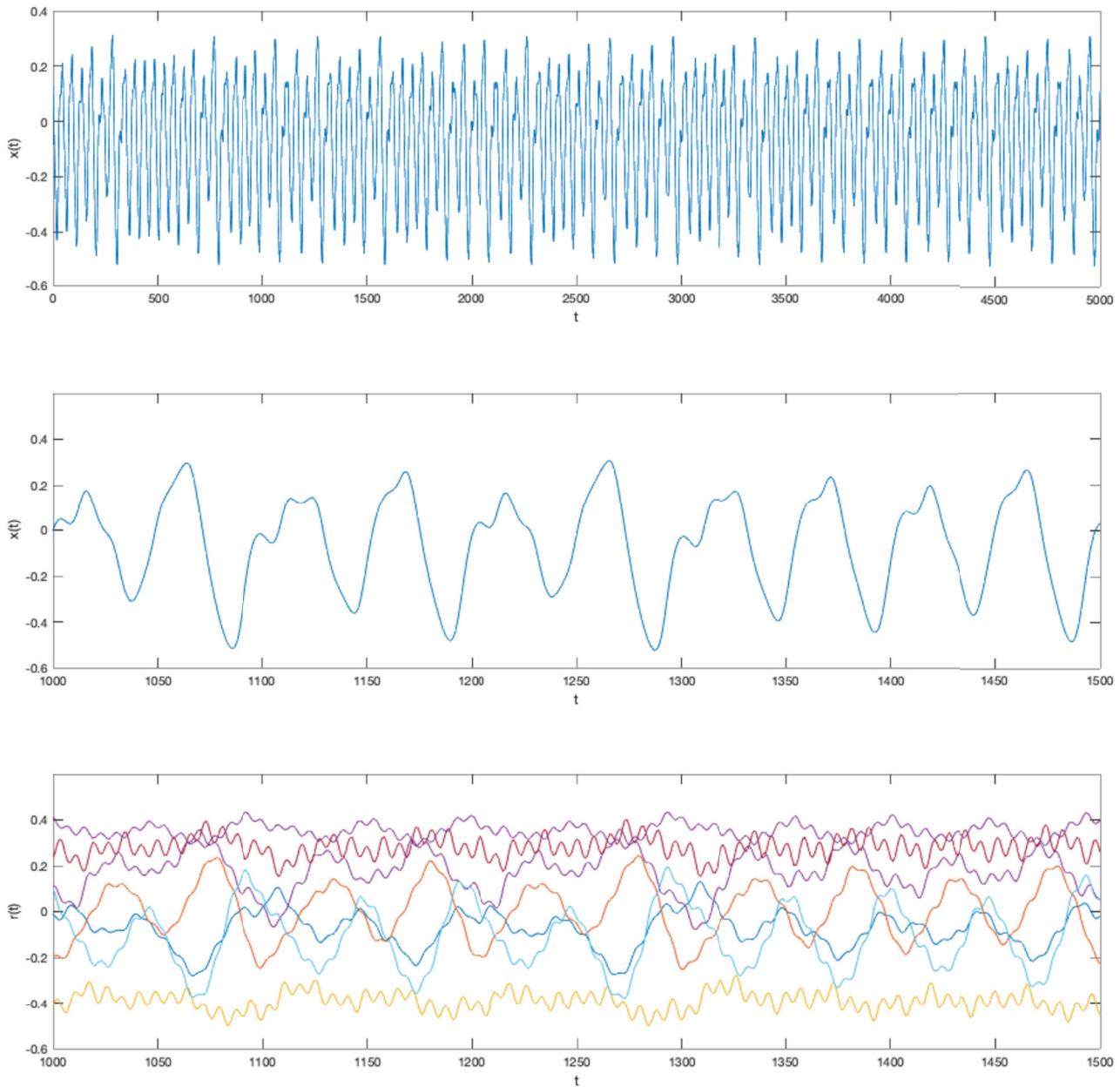


FIG. 3. Standard nonlinear RC, one-time-step forecasts from the Mackey–Glass differential delay equation [Eq. (60)], using a training data set from $N = 10\,000$ samples as shown in Fig. 1. (Top) Time-series data, $N = 5000$, shown for clearer illustration. (Middle) Reservoir trained across the data set, and 500 samples are shown for clarity where we see the error is sufficiently small that the one-time-step forecasts and the true data are almost the same so that the plot is indistinguishable (both shown, but curves overlap). Regularity is chosen to be $\lambda = 1 \times 10^{-6}$. (Bottom) Some randomly selected seven of the (usually hidden) $d_r = 500$ activation functions illustrate the general appearance. Contrast to forecasting into the future as shown in Fig. 4 and the linear method in Fig. 5.

quadratic readout; that is, extending Eq. (8) to also include terms $\mathbf{r} \circ \mathbf{r}$ (componentwise multiplication, “ \circ ” is called the Hadamard product) yields a quadratic NVAR of all monomial quadratic terms, which we observe performs quite well.

IV. RC WITH A FULLY LINEAR ACTIVATION, $q(s) = s$, YIELDS A VAR(k)

Now, we attempt to challenge a central typical assumption of the RC method. Instead of choosing the activation function to be

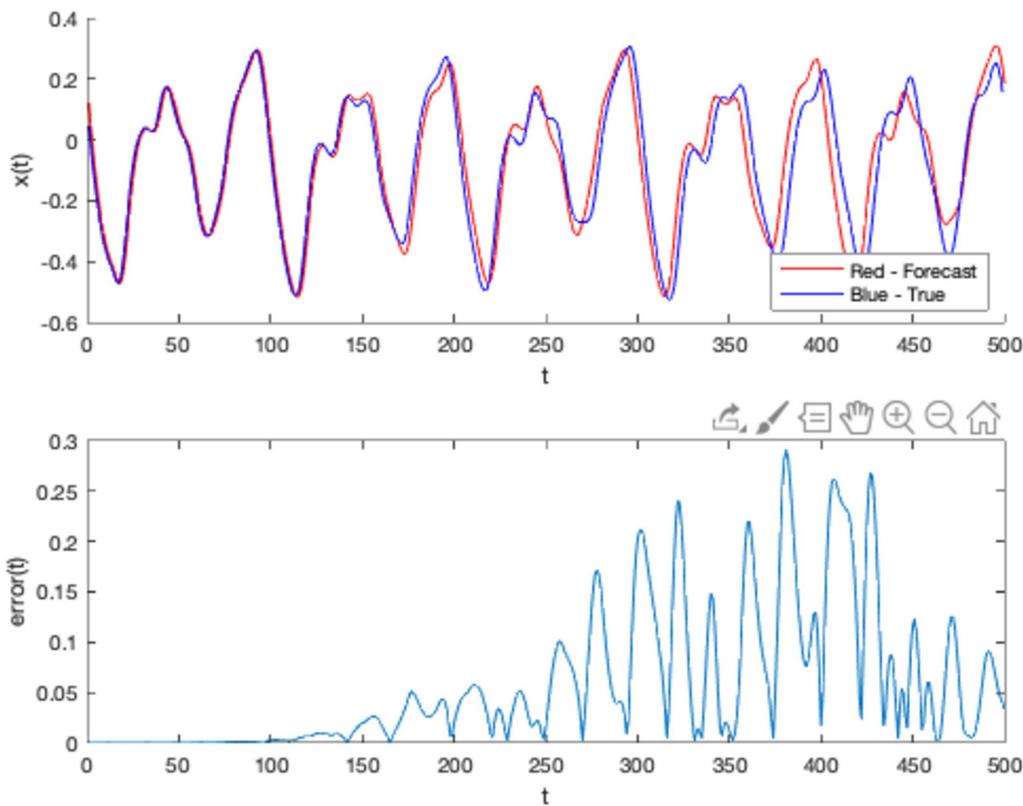


FIG. 4. Standard nonlinear RC, forecasts into the future, from the Mackey–Glass differential delay equation [Eq. (60)], using a training data set from $N = 10\,000$ samples as shown in Figs. 1 and 3. (Top) Time-series data, $0 < t < 500$, zoom plotted for clearer illustration. (Top) Forecasts into the future (red) diverge from true (blue), and (bottom) the error is shown. All forecasts shown will be closed-loop, once trained on the training set, meaning that the RC output is then input back into the RC input. We compare the error vs the true evolution, as is standard such as Ref. 56.

a sigmoid function, we use the identity function, $q(x) = x$. With this assumption, we can show that the resulting linear RC machine is equivalent to a vector autoregressive process (VAR),^{37,73} which is extremely popular and successful in the time-series forecasting field, particularly in econometrics.¹ With this simplification, we find that not only can the linear RC still make useful forecasts, but also, we are able to connect the RC concept to this well established theory associated with VAR time-series analysis, notably the existence of the representation Wold theorem.^{68,92} However, while this gives some explanation as to why a standard nonlinear RC may work despite the seemingly oversimplification of a full RNN, we show that the linear RC does still perform and furthermore, now with theoretical underpinnings, even if the full nonlinear RC may still perform better. Therefore, it is for the theoretical connections that we make this simplification rather than a suggestion that it may be a new or simpler method.

Before proceeding with a discussion of $q(s) = s$, notice that r is related to the scale of the read-in matrix, \mathbf{W}^{in} . Proceed by initializing the process, by Eq. (5),

$$\mathbf{u}_1 = \mathbf{W}^{in}\mathbf{x}_1, \text{ but also, we choose } \mathbf{r}_1 = 0. \quad (11)$$

Consider that since \mathbf{W}^{in} is randomly chosen and we choose uniformly $\mathbf{W}^{in} \sim U(0, \gamma)$, then the parameter $\gamma > 0$ moderates the subsequent scale of terms \mathbf{u}_i and then \mathbf{r}_i . See, for example, Fig. 3, where the native data \mathbf{x} from the Mackey–Glass system are translated to scaled internal variables. Recall the power series of the nonlinear activation function,

$$q(s) = \tanh(s) \approx s - s^3/3 + s^5/5 - \dots. \quad (12)$$

Clearly, for $s \ll 1$, then $q(s) \sim s$ even if chosen as a sigmoid, and the choice of read-in scale could be designed to put us in this regime as long as \mathbf{A} is designed to keep us in this regime. That is, if we choose the scale of the read-in matrix, $0 < \gamma \ll 1$, giving small values of the matrix \mathbf{W}_{ij}^{out} , then at least for a stable RC such as when \mathbf{A} has a sufficiently small spectral radius, the arguments of s from $\mathbf{Ar} + \mathbf{u} + \mathbf{b}$ in Eq. (4) remain small. Therefore, in practice, $\tanh(s) \sim s$. Stated roughly, if γ is small, then at least for some short time, we might expect that the fully nonlinear RC is close to a fully linear RC. To advance beyond that as an idea, for now, we believe the insights gained for a linear activation RC should be relevant to the general problem.

In the following, we proceed to study the consequences of stating the activation exactly as the identity,

$$q(s) = s. \quad (13)$$

With this assumption, the first several iterations follow from Eq. (4) and Eq. (11) as a forward propagation for which we explicitly observe the following recursion:

$$\mathbf{r}_2 = \mathbf{A}\mathbf{r}_1 + \mathbf{u}_1 = \mathbf{u}_1 = \mathbf{W}^{in}\mathbf{x}_1, \quad (14)$$

$$\begin{aligned} \mathbf{r}_3 &= \mathbf{A}\mathbf{r}_2 + \mathbf{u}_2 \\ &= \mathbf{A}\mathbf{W}^{in}\mathbf{x}_1 + \mathbf{W}^{in}\mathbf{x}_2, \end{aligned} \quad (15)$$

$$\begin{aligned} \mathbf{r}_4 &= \mathbf{A}\mathbf{r}_3 + \mathbf{u}_3 \\ &= \mathbf{A}(\mathbf{A}\mathbf{r}_2 + \mathbf{u}_2) + \mathbf{u}_3 \\ &= \mathbf{A}^2\mathbf{W}^{in}\mathbf{x}_1 + \mathbf{A}\mathbf{W}^{in}\mathbf{x}_2 + \mathbf{W}^{in}\mathbf{x}_3, \end{aligned} \quad (16)$$

⋮

$$\begin{aligned} \mathbf{r}_{k+1} &= \mathbf{A}\mathbf{r}_k + \mathbf{u}_k \\ &= \mathbf{A}(\mathbf{A}\mathbf{r}_{k-1} + \mathbf{u}_{k-1}) + \mathbf{u}_k, \\ &\vdots \\ &= \mathbf{A}^{k-1}\mathbf{W}^{in}\mathbf{x}_1 + \mathbf{A}^{k-2}\mathbf{W}^{in}\mathbf{x}_2 + \cdots + \mathbf{A}\mathbf{W}^{in}\mathbf{x}_{k-1} + \mathbf{W}^{in}\mathbf{x}_k \end{aligned} \quad (17)$$

$$= \sum_{j=1}^k \mathbf{A}^{j-1}\mathbf{u}_{k-j+1} = \sum_{j=1}^k \mathbf{A}^{j-1}\mathbf{W}^{in}\mathbf{x}_{k-j+1}, \quad (18)$$

using notation, $\mathbf{A}^0 = \mathbf{I}$, the identity matrix. Since the readout of this process is by Eq. (4), $\mathbf{y}_i = \mathbf{W}^{out}\mathbf{r}_i$, then we may rewrite the final equation [Eq. (18)] by multiplying the left-hand side by \mathbf{W}^{out} ,

$$\begin{aligned} \mathbf{y}_{\ell+1} &= \mathbf{W}^{out}\mathbf{r}_{\ell+1} \\ &= \mathbf{W}^{out} \sum_{j=1}^{\ell} \mathbf{A}^{j-1}\mathbf{W}^{in}\mathbf{x}_{\ell-j+1} \\ &= \mathbf{W}^{out}\mathbf{A}^{\ell-1}\mathbf{W}^{in}\mathbf{x}_1 + \mathbf{W}^{out}\mathbf{A}^{\ell-2}\mathbf{W}^{in}\mathbf{x}_2 + \cdots \\ &\quad + \mathbf{W}^{out}\mathbf{A}\mathbf{W}^{in}\mathbf{x}_{\ell-1} + \mathbf{W}^{out}\mathbf{W}^{in}\mathbf{x}_{\ell} \\ &= a_{\ell}\mathbf{x}_1 + a_{\ell-1}\mathbf{x}_2 + \cdots + a_2\mathbf{x}_{\ell-1} + a_1\mathbf{x}_{\ell}, \end{aligned} \quad (19)$$

with notation

$$a_j = \mathbf{W}^{out}\mathbf{A}^{j-1}\mathbf{W}^{in}, j = 1, 2, \dots, \ell. \quad (20)$$

Each of these coefficients a_j are $d_x \times d_x$ matrices. This follows simply by Eq. (20), collecting products between $d_x \times d_r$ to $d_r \times d_r$ and then $d_r \times d_x$ matrices and notation $\mathbf{A}^l = \prod_{i=1}^l \mathbf{A} = \mathbf{A} \cdot \mathbf{A} \cdots \mathbf{A}$, l -times if $l > 0$ or the identity matrix when $l = 0$.

In some sense, Eq. (20) and quadratic generalization equations (52) and (53) are the heart of this paper as it is an explicit representation of the coefficient matrices of a VAR, (or NVAR), but as found in terms of projection onto iterations of the random matrices involved in developing a linear activation function version of an RC. With exactly k VAR matrices a_j , the randomness of the d_r^2 free parameters of the random matrix \mathbf{A} collapses onto kd_x^2 parameters,

meaning that it yields only the many finitely fitted parameters of the matrices of a_1, \dots, a_k . However, for longer observations, which are the more usual way an RC is trained and practiced, Eq. (20) implies that when condition

$$d_r < kd_x, \quad (21)$$

the randomness of the choice of \mathbf{A} and \mathbf{W}^{in} is completely specified by stating matrices $\mathbf{A}^{j-1}\mathbf{W}^{in}$, for many j . We will expand upon this statement in Sec. V and then show how it relates to vanishing memory in Sec. IX.

By Eq. (19), a linear RC yields a classical VAR(k) (a vector autoregression model of k -delays) that in a general form is⁷³

$$\mathbf{y}_{k+1} = c + a_k\mathbf{x}_1 + a_{k-1}\mathbf{x}_2 + \cdots + a_2\mathbf{x}_{k-1} + a_1\mathbf{x}_k + \xi_{k+1}. \quad (22)$$

In this writing, c allows for a general offset term, a $d_x \times 1$ vector that here we do not pursue. ξ_{k+1} is the underlying “noise” of the stochastic process, which is part of the stability theory that we review in Sec. V, which must be assumed to come from a covariance stationary process. This relationship between an RC and a VAR(k) allows us to relate to the corresponding theoretical discussions of relevant alternative forms and stability and convergence from the stochastic process time-series literature, which we will also expand upon in Sec. V.

Considering the complete data set of vector time-series, $\{\mathbf{x}_i\}_{i=1}^N$, yields

$$\begin{bmatrix} | & | & | & | \\ \mathbf{y}_{k+1} & \mathbf{y}_{k+2} & \dots & \mathbf{y}_N \\ | & | & | & | \end{bmatrix} = \begin{bmatrix} [a_1] & [a_2] & \dots & [a_k] \end{bmatrix}$$

$$\times \begin{bmatrix} | & | & \vdots & | \\ \mathbf{x}_k & \mathbf{x}_{k+1} & \dots & \mathbf{x}_{N-1} \\ | & | & \vdots & | \\ \mathbf{x}_{k-1} & \mathbf{x}_k & \dots & \mathbf{x}_{N-2} \\ | & | & \vdots & | \\ \vdots & \vdots & \vdots & \vdots \\ | & | & \vdots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_{N-k} \\ | & | & \vdots & | \end{bmatrix}. \quad (23)$$

Restating this as a single linear equation,

$$\mathbf{Y} = \mathbf{a}\mathbb{X}. \quad (24)$$

Again, remembering that \mathbf{x}_i are $d_x \times 1$ vectors and that a_i are $d_x \times d_x$ matrices, $\mathbf{a} = [[a_1] | [a_2] | \dots | [a_k]]$ is a $d_x \times (kd_x)$ matrix. $\mathbf{Y} = [\mathbf{y}_{k+1} | \mathbf{y}_{k+2} | \dots | \mathbf{y}_N]$ is a $d_x \times (N - k)$ matrix, and \mathbb{X} is a $(kd_x) \times (N - k)$ matrix.

Formally, minimizing in least squares, with regularization,

$$J(\mathbf{a}) = \|\mathbf{Y} - \mathbf{a}\mathbb{X}\|_F + \lambda\|\mathbf{a}\|_F, \quad (25)$$

with \mathbf{Y} being the target output of the right hand side of Eq. (23) by best fitted matrix \mathbf{a}^* . The solution of this regularized least squares

problem may be written in its matrix form,

$$\mathbf{a}^* = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda I)^{-1} := \mathbf{Y}\mathbf{X}_\lambda^\dagger, \quad (26)$$

where the symbol \dagger refers to the Penrose pseudo-inverse, with the notation described in detail in Eqs. (A5) and (A6), when formulating the “ridge” Tikhonov regularized pseudo-inverse $\mathbf{X}_\lambda^\dagger$.

A. Decomposing the VAR(k) solution explicitly relates to RC

Now, we will further decompose the derived VAR(k) coefficients found in Eq. (26) to emphasize the training of just the output matrix \mathbf{W}_{out} of an associated RC in terms of randomly pre-choosing \mathbf{A} and \mathbf{W}^{in} .

Referring to Eqs. (19) and (20), we can rewrite Eqs. (23) and (24) as

$$\mathbf{Y} = \mathbf{a}\mathbf{X} = \mathbf{v}\mathbf{AX}, \quad (27)$$

with the matrix defined,

$$\mathbb{A} = [\mathbf{W}^{in} | \mathbf{AW}^{in} | \dots | \mathbf{A}^{k-2}\mathbf{W}^{in} | \mathbf{A}^{k-1}\mathbf{W}^{in}]. \quad (28)$$

This \mathbb{A} is a combination of exponents of the random $d_r \times d_r$ matrix \mathbf{A} and the random $d_r \times d_x$ matrix \mathbf{W}^{in} ; therefore, it is itself a $d_r \times kd_x$ random matrix. Interestingly, considering just one column at a time of the \mathbf{W}_l^{in} , $l = 1, 2, \dots, d_r$, \mathbf{A}^{k-1} can be understood as a collection of columns from a Krylov space, and this entire process can be discussed as an Arnoldi-iteration, which is something we will explore further in Sec. VII.

Consider that the least squares objective equation (30) can be expanded to split

$$\mathbf{a} = \mathbf{v}\mathbb{A} \quad (29)$$

to emphasize that since if we pre-choose \mathbf{A} and \mathbf{W}^{in} , then only the readout matrix \mathbf{v} is a free parameter,

$$J(\mathbf{v}) = \|\mathbf{Y} - \mathbf{a}\mathbf{X}\|_F = \|\mathbf{Y} - \mathbf{v}\mathbb{A}\mathbf{X}\|_F. \quad (30)$$

Optimizing for \mathbf{v} yields

$$\mathbf{W}^{out} := \mathbf{v}^* = \mathbf{Y}(\mathbb{A}\mathbf{X})^\dagger = (\mathbf{Y}\mathbf{X}^\dagger)\mathbb{A}^\dagger. \quad (31)$$

Comparing this equation with Eq. (26), defining \mathbf{a} , we see $(\mathbf{Y}\mathbf{X}^\dagger)$ formally appears in both expressions. Only the associative property of matrix multiplication is needed to emphasize the role of \mathbb{A} . More importantly, this expression [Eq. (31)] for \mathbf{W}^{out} is written so as to emphasize that the reservoir computing process is designed with \mathbb{A} and \mathbf{X} . Combined through the iteration, as $(\mathbb{A}\mathbf{X})$ is the data that result from Eq. (17),

$$\mathbf{r}_{\ell+1} = \mathbf{A}^{\ell-1}\mathbf{W}^{in}\mathbf{x}_1 + \mathbf{A}^{\ell-2}\mathbf{W}^{in}\mathbf{x}_2 + \dots + \mathbf{A}\mathbf{W}^{in}\mathbf{x}_{\ell-1} + \mathbf{W}^{in}\mathbf{x}_\ell. \quad (32)$$

This is written naturally as

$$\mathbf{R} = (\mathbb{A}\mathbf{X}). \quad (33)$$

In a simple way, Eq. (31) uses a matrix identity of pseudo-inverses,³¹

$$(\mathbb{A}\mathbf{X})^\dagger = \mathbf{X}^\dagger\mathbb{A}^\dagger. \quad (34)$$

Associativity emphasizes that since \mathbb{A} is deterministically defined, once \mathbf{A} and \mathbf{W}^{in} are chosen, and separately from the data \mathbf{X} , the

fitting of only the parameters of \mathbf{W}^{out} is sufficient. If we want the VAR(k) parameters, then we could either ignore the prior knowledge of choice of \mathbf{A} and \mathbf{W}^{in} and compute \mathbf{a} directly from Eq. (30) or from Eq. (31) by defining

$$\mathbf{W}^{out} := \mathbf{v}^* = \mathbf{a}^*\mathbb{A}_\lambda^\dagger = \mathbf{Y}\mathbf{X}_\lambda^\dagger\mathbb{A}_\lambda^\dagger. \quad (35)$$

We summarize that these manipulations concluding with Eq. (35) serve directly as the connection between the RC fitted readout and the coefficient matrices of a VAR(k). The roles of pre-choosing \mathbf{A} and \mathbf{W}^{in} relate directly to \mathbf{W}^{out} coefficients or indirectly to the fitted data. Considering the training of \mathbf{W}^{out} , Eqs. (29), (31), and (35), in terms of geometric description of least squares estimation,³¹ \mathbf{W}^{out} best estimates orthogonal projections of rows of \mathbf{a} into the row space of \mathbb{A} , which has d_r -row vectors of dimension kd_x . Therefore, no more than d_r dimensions can remain free or as described similarly in the inequality equation (21).

Concluding this section with an example, we simplify the nonlinear RC of the Mackey–Glass data from Figs. 1 and 3 to a purely linear RC fit shown in Fig. 5, which clearly is not as well performing, but it does still make some forecast into the future. Further discussion of this example and also likewise a Lorenz63 example is discussed in Sec. VIII.

V. VAR(k) THEORY SUGGESTS CONVERGENCE WITH k

Since the VAR(k) model of vector autoregression appears naturally in our discussion from the simplified activation function $q(x) = x$, as summarized by Eqs. (19) and (22), we now recall some of the classical underlying theories from the statistical time-series analysis literature^{73,92} that describe sufficient conditions under which we expect the existence of a VAR(k) representation.

The Wold theorem plays a central role in time-series analysis as it describes the existence of a vector moving average (VMA) model representation, which then under further assumption for invertibility is equivalent to a VAR. Assumptions require a stationary process as the sum of two components: (1) a stochastic component consisting of “linear” combinations of lags from a white noise process and (2) a deterministic component that is uncorrelated with the stochastic component. First, we recall the definitions. A d-dimensional stochastic process ξ_t of zero mean, $\mathbb{E}(\xi_t) = \mathbf{0}$, is derived from a white noise stochastic process, written with zero mean $\xi_t = [\xi_{1,t}, \xi_{2,t}, \dots, \xi_{d,t}] \sim WN(0, \Omega)$ if $\mathbb{E}(\xi_t) = \mathbf{0}$ and $\mathbb{E}(\xi_{t_1}\xi_{t_2}^T) = \mathbf{0}$, for $t_1 \neq t_2$, but $\mathbb{E}(\xi_t\xi_t^T) = \Omega$ is symmetric positive semi-definite. A stochastic process is covariance stationary if all terms of the sequence have the same mean and any two terms depend only on their relative positions. That is, $\mathbb{E}(\xi_{t'}) = \mathbb{E}(\xi_t)$, for all t' , and for all $t' \geq 0$, there exists $\gamma_{t'} \in \mathbb{R}$ such that $Cov(\xi_t, \xi_{t-t'}) = \gamma_{t'}$ for all $t > t'$, meaning depending on $t - t'$ rather than t or t' . With these definitions, we can state the central theorem of this section that we recall.

Theorem 1 (Wold decomposititon theorem^{73,92}): A zero mean covariance stationary vector process $\{\mathbf{x}_t\}$ admits a representation,

$$\mathbf{x}_t = C(L)\xi_t + \mu_t, \quad (36)$$

where $C(L) = \sum_{i=0}^{\infty} C_i L^i$ is a polynomial delay operator polynomial, C_i are the moving average matrices, and $L^i(\xi_t) = \xi_{t-i}$. The term

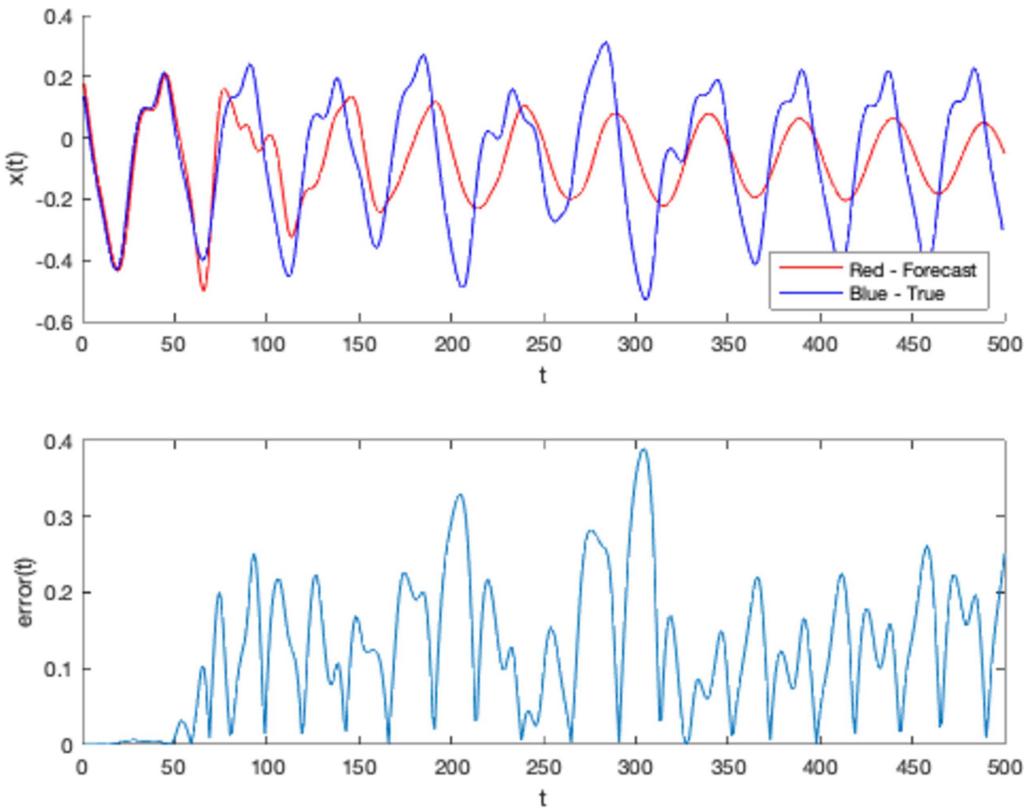


FIG. 5. The fully linear RC, $q(s) = s$, forecasts from the Mackey–Glass differential delay equation [Eq. (60)], using the same training data set from $N = 10\,000$ samples as shown in Fig. 1. Contrasting to forecasts into the future as shown in Fig. 4, we see that clearly, the nonlinear RC outperforms the linear RC and by a wide margin. However, that is not the message here, rather which is one of explaining the relationships and fitting of the parameters; therefore, that fitting just the readout matrix \mathbf{W}^{out} is relevant is established by Eq. (35).

$C(L)\xi$ is the stochastic part of the decomposition. The μ_t term is the deterministic (perfectly predictable) part as a linear combination of the past values of \mathbf{X}_t . Furthermore,

- μ_t is a d -dimensional linearly deterministic process.
- $\xi_t \sim WN(0, \Omega)$ is white noise.
- Coefficient matrices are square summable,

$$\sum_{i=0}^{\infty} \|C_i\|^2 < \infty. \quad (37)$$

- $C_0 = I$ is the identity matrix.
- For each t , μ_t is called the innovation or the linear forecast errors.

Clarifying notation of the delay operator polynomial, with an example, let

$$C(L) = \begin{bmatrix} 1 & 1+L \\ -\frac{1}{2}L & \frac{1}{2}-L \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & \frac{1}{2} \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -\frac{1}{2} & -1 \end{bmatrix}$$

$$L = C_0 + C_1 L, \text{ and } C_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ if } i > 1; \quad (38)$$

therefore if, for example, $\mathbf{x}_t \in \mathbb{R}^2$,

$$C(L)\mathbf{x}_t = \begin{bmatrix} 1 & 1+L \\ -\frac{1}{2}L & \frac{1}{2}-L \end{bmatrix} \begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} = \begin{bmatrix} x_{1,t} + x_{2,t} + x_{2,(t-1)} \\ \frac{1}{2}x_{1,(t-1)} + \frac{1}{2}x_{2,t} - x_{2,(t-1)} \end{bmatrix}. \quad (39)$$

For interpretation and definition, consider

- If $\mu_t = 0$, then this is called a “regular” process; therefore, there is a purely vector moving average (VMA) representation. If $C_i = 0$ for $i > p$ for some finite $p > 0$, then it is called a VMA(p) or otherwise, it is a VMA(∞) representation.
- If \mathbf{X}_t is regular, then the representation is unique.

Now, we consider our point to relate a Wold VMA representation to the linear RC where we already saw a VAR(k) results, [Eq. (19)], when the activation is linear $q(x) = x$. If the delay polynomial $C(L)$ is invertible with $C(L)^{-1}C(L) = I$ and denotes $C(L)^{-1} = B(L) = B(L) = B_0 - B_1 L - B_2 L^2 - \dots$ in terms of matrices B_i , then writing explicitly,

$$(B_0 - B_1 L - B_2 L^2 - \dots)(I + C_1 L + C_2 L^2 + \dots) = I. \quad (40)$$

The existence of this inverse implies that the Wold implied VMA process has a representation,

$$\mathbf{X}_t = C(L)\xi_t \implies B(L)\mathbf{X}_t = \xi_t, \quad (41)$$

which is a VAR representation in that this represents the latest \mathbf{X}_t as a linear combination of prior values of \mathbf{X} written succinctly in terms of the delay operator $B(L)$.

In practice, when an infinite order vector moving average process $VMA(\infty)$ corresponds to an infinite order vector autoregressive process, $VAR(\infty)$, then recursion of expanding Eq. (40) and matching term by term yields

$$B_0 = I, B_1 = C_1, \dots, B_k = C_k + B_1 C_{k-1} + \dots + B_{k-1} C_1, \dots \quad (42)$$

However, a VAR representation may be found from a VMA through several methods, including a method of moments leading to the Walker–Yule equations⁷¹ or a least squares method in the case of finite presentations. Often, for parsimonious efficiency of presentation, a mixed form of a p -step AR and a q -step MA model might make a suitable approximation for what is called a ARMA(p,q) model.

While not allowing ourselves to be drawn entirely into the detailed theory of econometrics and statistical time-series analysis, pursuing stronger necessary conditions, we wish to point out some already apparent relevant points from the stated special sufficient conditions.

Remark 1: *Summary statements. If a vector stochastic process satisfies the hypothesis of a Wold theorem, then*

- it can be written either as a VMA or a VAR, when Eq. (41) of $C(L)$ is invertible [Eq. (42)];
- in practice, a finite k , $VAR(k)$ estimates a $VAR(\infty)$ as $k \uparrow$, since the sequence of coefficient matrices $\{C_i\}$ is square summable [Eq. (37)] and considering Eq. (42); and
- furthermore, in practice, a least squares estimate of a $VAR(k)$ may be used for finite k , which relates to an RC by the least squares fit [Eqs. (31) and (35)].

Finally, we separate from the above technical points the following fundamental remark to distinguish existence vs uniqueness of a representation,

Remark 2: *While a stochastic process may have a VMA representation and if through invertibility, a corresponding VAR, which is a linear description of the process, it may not taken to be “the” unique physical underlying description since nonlinear descriptions certainly may exist.*

Remark 3: *The processes that we may be interested in, such as those derived from Eq. (1), may describe the evolution of a (chaotic) dynamical system, and these may allow a representation [Eq. (3)].^{10,12,51,82} However, in many of these natural examples, the “color” or even the nature of the noise may well not be conforming to the white noise assumption of the Wold Theorem 1. Certainly, contrasting samples from an invariant measure of a chaotic dynamical system to a white noise process is a well studied^{46,75} but still undecided topic. While the existence of the VMA and the corresponding VAR representation by referring to the Wold theorem does depend on that hypothesis, nonetheless, successful constructive fitting of a $VAR(k)$ by regression, even if implicitly through an RC, seems to proceed successfully in practice in a wide array of examples.*

With this last remark, we admit that while the details of the rigor guaranteeing existence may in practice break down, due to inability to check all hypotheses, as often such gaps occur between mathematics, applied mathematics, and practice as related to real world data, we feel that the concept is still highly instructive as underlying explanation despite strong sufficient assumptions used to extend the rigorous theory.

We summarize this section that the relationship between the Wold theorem for the VAR to our interest in an RC gives two conclusions: (1) the existence of the VMA representation follows the Wold, which in turn leads to a VAR when the delay operator is invertible and (2) that the coefficient matrices are square summable serves as an upper bound that memory must be fading. We describe the memory further in Sec. IX.

A. Stability of the $VAR(k)$ and relationship to a $VAR(1)$

To discuss stability, we recall⁷³ the fact that a $VAR(k)$ [Eqs. (19) and (22)],

$$\mathbf{x}_{k+1} = \mathbf{c} + a_k \mathbf{x}_1 + a_{k-1} \mathbf{x}_2 + \dots + a_2 \mathbf{x}_{k-1} + a_1 \mathbf{x}_k + \xi_{k+1}, \quad (43)$$

can be stated as a $VAR(1)$ in terms of “stacked” (delayed) variables, called the companion system. This idea is familiar in dynamical systems as we see it is related to stating time-delay variables and Taken’s embedding theorem.^{9,64,67,77,83,95} Define

$$\mathbb{X}_{k+1} = \mathcal{A}\mathbb{X}_k + \mathcal{C} + \mathbf{e}_k, \text{ where } \mathbb{X}_k = \begin{bmatrix} | & | & | \\ \mathbf{x}_k & \mathbf{x}_{k-1} & \vdots \\ | & | & | \end{bmatrix}, \mathbf{e}_k = \begin{bmatrix} | & | \\ \xi_k & 0 \\ | & | \\ \vdots & \vdots \\ | & | \\ \mathbf{x}_1 & 0 \\ | & | \end{bmatrix},$$

$$\mathcal{C} = \begin{bmatrix} | & | & | \\ \mathbf{c} & 0 & \vdots \\ | & | & | \end{bmatrix}, \text{ and } \mathcal{A} = \begin{bmatrix} a_1 & a_2 & \cdots & a_k \\ I & 0 & \cdots & 0 \\ 0 & I & \ddots & 0 \\ 0 & \cdots & I & 0 \end{bmatrix}, \quad (44)$$

where since a_i are each $d_x \times d_x$ matrices, \mathcal{A} is $kd_x \times kd_x$ and \mathbb{X}_k is $kd_x \times 1$. For discussion in Sec. VI, it will be convenient to consider in contrast to Eq. (54), a matrix of all the data,

$$\mathbb{X} = [\mathbb{X}_k \ \mathbb{X}_{k+1} \ \cdots \ \mathbb{X}_{N-k-1}]; \text{ likewise let} \\ \mathbb{X}' = [\mathbb{X}_{k+1} \ \mathbb{X}_{k+2} \ \cdots \ \mathbb{X}_{N-k}], \quad (45)$$

$kd_x \times (N - k - 1)$. Notice that the data in \mathbb{X} are also from Eq. (23).

It follows that analysis of stability of a $VAR(1)$ sufficiently describes the stability of a $VAR(k)$. If there is even a small offset \mathbf{c} ,

whether by a bias or imperfection of fit, then it follows the recursion,

$$\begin{aligned} \mathbb{X}_k &= \mathcal{C} + \mathcal{A}\mathbb{X}_{k-1} + \mathbf{e}_{k-1}, \implies \mathbb{X}_k = (I + \mathcal{A} + \cdots + \mathcal{A}^{l-1})\mathcal{C} \\ &\quad + A^l\mathbb{X}_{k-l} + \sum_{j=0}^{l-1} \mathcal{A}^j \mathbf{e}_{k-l}. \end{aligned} \quad (46)$$

This relates the VAR(1) back to a VMA(l) form. Clearly, even a small constant disturbance \mathcal{C} is successively influenced by the (delay) matrix \mathcal{A} . In the limit $l \rightarrow \infty$, recall the geometric series of matrices,

$$(I - \mathcal{A})^{-1} = \lim_{l \rightarrow \infty} (I + \mathcal{A} + \cdots + \mathcal{A}^{l-1}), \quad (47)$$

converges if the spectral radius is strictly contained in the complex unit disk,

$$\rho(\mathcal{A}) = \max_{\lambda: \det(A - \lambda I) = 0} |\lambda| < 1. \quad (48)$$

Equivalently, a general VAR(k) [Eq. (43)] is stable if and only if a characteristic polynomial,

$$\det(I - a_1 z - a_2 z^2 - \cdots - a_k z^k) = 0, \quad (49)$$

has all its roots outside the unit disk. Under this stability assumption, we conclude that

$$\mathbb{X}_k = \mathcal{C} + \mathcal{A}\mathbb{X}_{k-1} + \mathbf{e}_{k-1} = (I - \mathcal{A})^{-1}\mathcal{C} + \sum_{j=0}^{\infty} \mathcal{A}^j \mathbf{e}_{k-l}, \quad (50)$$

which relates a VAR(1) form to a Wold form through a VMA(∞). Since by Eq. (20), each matrix $a_j = \mathbf{W}^{out} \mathbf{A}^{j-1} \mathbf{W}^{in}$, the magnitude of entries in the matrix \mathbf{A} and the read-in matrix \mathbf{W}^{in} each moderate the magnitudes of entries of a_j . Therefore, considerations by the Gershgorin disk theorem³¹ relates these magnitudes to the magnitudes of z . Generally, by controlling the sparsity of \mathbf{A} , the magnitude of the spectrum of \mathbf{A} , and the magnitudes of \mathbf{W}^{in} , stability and the “memory” associated with converges (with k), can be controlled. These magnitudes were already discussed for the sake of a regime where the usual sigmoidal q would be close to the identity.

VI. QUADRATIC NONLINEAR VAR

In this section, as is common practice,⁷⁰ we investigate an RC that fits the readout \mathbf{W}^{out} using not just linear \mathbf{r} values from the RC but also in terms of $\mathbf{r} \circ \mathbf{r}$. Notation “ \circ ” is the Hadamard product, meaning componentwise multiplication, $[\mathbf{r} \circ \mathbf{r}]_j = [\mathbf{r}]_j^2$ for each j . This yields \mathbf{W}^{out} that is $d_x \times 2d_r$. Here, we state results briefly with detailed derivations given in Appendix B.

Generalizing the VAR results of Sec. IV, here, we state that a linear VAR with a Hadamard product-quadratic readout is equivalent to a quadratic nonlinear VAR, (NVAR) with all quadratic terms in $r_i r_j$. That is, analogously to the VAR stated equations (19) and (22), a quadratic nonlinear VAR may be stated [abbreviated restatement of Eqs. (B10)–(B12), derived in Appendix B],

$$\begin{aligned} \mathbf{y}_{k+1} &= a_k \mathbf{x}_1 + a_{k-1} \mathbf{x}_2 + \cdots + a_2 \mathbf{x}_{k-1} + a_1 \mathbf{x}_k + a_{2,(k,k)} p_2(\mathbf{x}_1, \mathbf{x}_1) \\ &\quad + a_{2,(k-1,k)} p_2(\mathbf{x}_2, \mathbf{x}_1) + \cdots + a_{2,(1,k)} p_2(\mathbf{x}_k, \mathbf{x}_1), \end{aligned} \quad (51)$$

with the notation for the k linear coefficient $d_x \times d_x$ matrices,

$$a_j = \mathbf{W}_1^{out} \mathbf{A}^{j-1} \mathbf{W}^{in}, j = 1, 2, \dots, k. \quad (52)$$

Now, we have k^2 quadratic term $d_x \times d_x^2$ coefficient matrices,

$$a_{2,(i,j)} = \mathbf{W}_2^{out} P_2(A^{i-1} \mathbf{W}^{in}, A^{j-1} \mathbf{W}^{in}), i, j = 1 \dots k. \quad (53)$$

The notation, $p_2(\mathbf{v}, \mathbf{w}) = [v_1 w_1 | v_1 w_2 | \cdots | v_n w_n]^T$ defines a n^2 -vector of all quadratic terms stated between vectors $\mathbf{v} = [v_1 | \cdots | v_n]^T$ and $\mathbf{w} = [w_1 | \cdots | w_n]^T$. $P_2(A^{i-1} \mathbf{W}^{in}, A^{j-1} \mathbf{W}^{in})$ is a $d_x \times d_x^2$ coefficient matrix built from columnwise Hadamard products. Both of these are expanded upon further in Eqs. (B2)–(B5), and the form of $a_{2,(i,j)}$, Eq. (53), is derived in Eq. (B12).

We summarize in this brief section that the discussion of nonlinear quadratic VAR from a linear RC with Hadamard quadratic readout is similar to that of linear VAR with linear RC and linear readout as discussed in Sec. IV. However, Eq. (35) generalizes so that we may still write $\mathbf{W}^{out} := \mathbf{v}^* = \mathbf{a}^* \mathbb{A}_\lambda^\dagger = \mathbf{Y} \mathbb{X}_\lambda^\dagger \mathbb{A}_\lambda^\dagger$, but now, $\mathbb{A} = \begin{bmatrix} \mathbb{A}_1 \\ \mathbb{A}_2 \end{bmatrix}$. In this statement, \mathbb{A}_1 is a renaming of what was \mathbb{A} in Eq. (28), but now, \mathbb{A}_2 is the $dr \times kd_x^2$ matrix defined in Eq. (B8) in Appendix B; likewise, $\mathbb{X} = \begin{bmatrix} \mathbb{X}_1 \\ \mathbb{X}_2 \end{bmatrix}$ is defined in Eq. (B9).

While the linear reservoir + linear readout = VAR forecasting seems to only give reasonable results for short time forecasting (Fig. 7), linear reservoir + quadratic readout = NVAR seems to give better and longer range forecasting, which also seems to remain true to the statistic of the chaotic attractor once errors have swamped the point forecasts (see Fig. 8).

VII. IS THERE A CONNECTION TO DMD-KOOPMAN?

To briefly answer the question titling this section, the answer is yes; there is a connection between VAR and DMD and so to RC. The more nuanced answer is that the connection is not complete. Throughout the discussion so far, a specialized version of an RC using an identity activation function yields a linear process that is shown to relate to a VAR that is also a linear process. In this section, we ask if it also relates to the dynamic mode decomposition (DMD),^{49,76,78,91} a concept that is also premised on a linear process model as a finite estimation of the infinite-dimensional linear action of the Koopman operator on a function space of observables.³ In the Koopman theory, instead of describing the evolution and geometry of orbits in the phase space, the transfer operator methods generally describe the evolution of functions whose domain is the phase space.^{12,51} Recently, this approach has excited a huge trend in applied dynamical systems, with many excellent research papers,^{8,49,53,91} review papers,^{3,16} and books⁴⁹ toward theory, numerical implementation, and scientific application practice. Our focus here will remain narrow, the goal being to simply identify a connection to the RC and its related VAR, as discussed above. The primary purpose of DMD methods is for modal analysis of the system to describe coherent and typical behaviors, but it also can be used for forecasting; for this reason, the analogy is drawn here.

For direct comparison, first allow some minor manipulations to relate the VAR(k) [Eqs. (43) and (23)] to a typical DMD form. A time-delay version of a linear evolution is a special case of an exact

DMD written as follows, with the notation used as above,

$$\begin{bmatrix} | & | & \vdots & | \\ \mathbf{x}_{k+1} & \mathbf{x}_{k+2} & \cdots & \mathbf{x}_N \\ | & | & \vdots & | \\ \mathbf{x}_k & \mathbf{x}_{k+1} & \cdots & \mathbf{x}_{N-1} \\ | & | & \vdots & | \\ \vdots & \vdots & \vdots & \vdots \\ | & | & \vdots & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{N-k} \\ | & | & \vdots & | \end{bmatrix} = \mathcal{K} \begin{bmatrix} | & | & \vdots & | \\ \mathbf{x}_k & \mathbf{x}_{k+1} & \cdots & \mathbf{x}_{N-1} \\ | & | & \vdots & | \\ \mathbf{x}_{k-1} & \mathbf{x}_k & \cdots & \mathbf{x}_{N-2} \\ | & | & \vdots & | \\ \vdots & \vdots & \vdots & \vdots \\ | & | & \vdots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{N-k-1} \\ | & | & \vdots & | \end{bmatrix}, \quad (54)$$

or simply,

$$\mathbb{X}' = \mathcal{K}\mathbb{X}, \quad (55)$$

where \mathbb{X} and \mathbb{X}' are the $kd_x \times (N - k - 1)$ data matrices in Eq. (45) and \mathcal{K} is a $kd_x \times kd_x$ DMD matrix approximating the action of the infinite-dimensional Koopman operator. Abusing notation slightly, the least squares problem,

$$\mathcal{K} = \arg \min_K \|\mathbb{X}' - K\mathbb{X}\|_F, \quad (56)$$

has the solution

$$\mathcal{K} = \mathbb{X}'\mathbb{X}^\dagger, \quad (57)$$

which is called the “exact DMD” solution. While there are many variants of DMD, this so-called exact DMD is popular for its simplicity of implementation while still being useful for interpreting the system in terms of modal behaviors.

Contrasting \mathcal{K} derived by exact DMD [Eq. (54)] vs \mathcal{A} for the VAR(1) form described in Eqs. (44) and (45) reveals clear similarities since each states a linear relationship between the same data, $\mathbb{X}' = \mathcal{K}\mathbb{X}$ vs $\mathbb{X}' = \mathcal{A}\mathbb{X}$, but these are ill-posed equations and \mathcal{A} need not be the same as \mathcal{K} . Closer inspection reveals that Eq. (57) allows freedom for a best least squares fit considering the entire matrix \mathcal{K} ; therefore, differences are relative to Eqs. (23) and (26). Whereas only the first k rows of \mathcal{A} are free parameters in the regression, the subsequent rows of \mathcal{A} are sparsely patterned with either zeros or the identity matrix [Eq. (44)].

A similar, but not identical, structural difference appears when contrasting the SVD based exact DMD to the original DMD method of Schmid⁷⁸ and also Rowley and Mezić,⁷⁶ which is an Arnoldi-like version of DMD in terms of iterations in a Krylov space.^{4,85} Reviewing that Arnoldi-version of DMD, using the notation of Ref. 76, observations $\mathbf{x}_k \in \mathbb{R}^d$ are assumed (fitted) to be from a linear process and also by considering the iterations are to be fitted in the Krylov space, assuming that $\mathbf{x}_m \in \text{Kry}_m(\mathbf{x}_0) = \text{span}\{\mathbf{x}_0, A\mathbf{x}_0, \dots, A^{m-1}\mathbf{x}_0\}$ for data, $K = [\mathbf{x}_0 | \mathbf{x}_1 | \dots | \mathbf{x}_m] = [\mathbf{x}_0 | A\mathbf{x}_0 | \dots | A^{m-1}\mathbf{x}_0]$. The linear combination $\mathbf{x}_m = A\mathbf{x}_{m-1} = c_0\mathbf{x}_0 + \dots + c_{m-1}\mathbf{x}_{m-1} = \mathbf{K}\mathbf{c}$, where $\mathbf{c} = [c_0; c_1; \dots; c_{m-1}]$ is the vector of coefficients. Then, a key and clever

observation was to rewrite this in terms of a companion matrix,

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & c_1 \\ 1 & 0 & & 0 & c_1 \\ 0 & 1 & & 0 & c_2 \\ \vdots & \ddots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & c_{m-1} \end{bmatrix}. \quad (58)$$

Therefore, the results are

$$AK = KC. \quad (59)$$

From there, exploiting the theme of Arnoldi methods, the eigenvalues of C are related as a subset of the eigenvalues of A , and with a direct linear relationship between eigenvectors of C and A , Ritz vectors and the unknown coefficients \mathbf{c} of C can be computed by a least squares procedure. Keeping in mind that power iterations as one does in Krylov spaces emphasize just the dominant direction, the Arnoldi methods take care to orthogonalize at each step in the algorithm for stabilization an otherwise unstable search for large sparse matrices, and these make deliberate use of QR decompositions. Our interest here is only to point out analogies between \mathcal{A} from reservoir computing and VAR(1) and \mathcal{K} rather than to continue toward discussion of modal analysis as one does in DMD analysis. Summarizing, the analogy we see is that the companion matrix C in Eq. (58) reminds us of the companion matrix \mathcal{A} in Eq. (44). However, the most significant difference is that while c_i are scalars, \mathbf{a}_i are $d_x \times d_x$ matrices.

VIII. EXAMPLES

The examples in Figs. 1 and 3–5 already threaded in the above presentation of methods were in terms of the Mackey–Glass differential delay equation system, which we now recall. Then, subsequently, we will show similar figures highlighting the concepts in a different system, the famous Lorenz63 ordinary differential equation.

A. Example 1: Mackey–Glass differential delay equation

The Mackey–Glass differential delay equation⁶¹

$$x'(t) = \frac{ax(t - t_d)}{1 + [x(t - t_d)]^c} - bx(t) \quad (60)$$

has now become a classic standard example in time-series analysis^{26,54} of a high (infinite)-dimensional dynamical system with a low-dimensional attractor, which we have used as a benchmark in our own previous work for machine learning⁹ dynamical systems. The problem is physiologically relevant for describing dynamic diseases. Differential delay equations can be described as infinite-dimensional dynamical systems, a concept that is more easily understandable in terms of the notion that an initial condition state advances to not just a finite-dimensional vector, but rather an entire interval $[t_0, t_0 + t_d]$ of initial values of $x(t)$ is required. However, the MG equations have a nice property for its practical use as a benchmark problem, which is that there is essentially an attractor whose fractal dimension varies with respect to the parameters chosen, allowing for a complexity tunable test problem. We have chosen

parameters $t_d = 17$, $a = 0.2$, $b = 0.1$, and $c = 10.0$ in which parsing time-delay embedding gives an embedding dimension of $d = 4$. We use integration steps of $\Delta t = t_d/100$ throughout. We show time-series in Fig. 1 a standard nonlinear RC forecast of the system in Fig. 3 and the linear RC/VAR forecast of the system in Figs. 4 and 5.

B. Example 2: Lorenz63

The Lorenz63 system⁵⁵ is the three coupled ordinary differential equations,

$$\begin{aligned}\dot{x} &= 10(y - x), \\ \dot{y} &= x(28 - z) - y, \\ \dot{z} &= xy - (8/3)z.\end{aligned}\quad (61)$$

While these Lorenz equations may have been originally posed as time varying Fourier coefficients for describing a partial differential equation system describing convection rolls of heated fluid in an atmospheric system, they have become a popular paradigm in the study of chaotic systems for foundation principles of chaos historically and ongoing, as a simple and familiar benchmark problem and also in the pedagogy of dynamical systems. The chaotic attractor in the phase space $\{x(t), y(t), z(t)\}$ illustrates a familiar butterfly, but we show a segment of the $x(t)$ time series that will be used as our data set in Figs. 6–8. Also shown are nonlinear RC, $q(x) = \tanh(x)$ activation forecasts in Fig. 6 using the usual nonlinear reservoir computing, with excellent success. In Fig. 7, we show forecasting results using a linear $q(x) = x$ activation RC with still good results for short-term forecasting agreeable with the expectation with the VAR theory.

Consider the following summary of results of three experiments shown in Figs. 6–8.

- Figure 6 shows a standard fully nonlinear RC, with activation $q(x) = \tanh(x)$ and Hadamard quadratic readout \mathbf{W}^{out} of the reservoir variable $[\mathbf{r}; \mathbf{r} \circ \mathbf{r}]$. Forecasting is excellent; interestingly, apparently, even once errors have accumulated, the RC continues to produce a Lorenz-like attractor.
- Figure 7 shows a fully linear RC with linear activation, $q(x) = x$, and linear readout \mathbf{W}^{out} of the reservoir variable \mathbf{r} is equivalent to a VAR and it produces good short-term forecasts, but for a shorter time than the fully nonlinear RC. Also, once errors accumulate, the long-term forecasts do not produce a Lorenz-like attractor but rather seems to converge to zero.
- Figure 8 shows a linear RC, so $q(x) = x$, but with a Hadamard quadratic readout \mathbf{W}^{out} of $[\mathbf{r}; \mathbf{r} \circ \mathbf{r}]$. Forecast performance is good and the attractor seems to be well reproduced even once the error has grown, comparably to the fully nonlinear case of Fig. 6.

IX. ON FADING MEMORY

There are several ways to consider memory in the system. Memory as it turns out is an important property for an echo-state machine since it was recently shown by Gonan and Ortega,³³ and that these are universal. The VAR representation shown here allows for an own discussion of this property.

A. Fading memory in terms of the role of the internal layers

One way is to think that the connection between the reservoir and the coefficient matrices \mathbf{a}_j may become increasingly small in a way that is moderated in part by the randomly chosen \mathbf{A} yielding a fading bound on the true memory. From Eq. (20), $a_j = \mathbf{W}^{out} \mathbf{A}^{j-1} \mathbf{W}^{in}$, follows a bound, in norm,

$$\begin{aligned}\|a_j\|_* &= \|\mathbf{W}^{out} \mathbf{A}^{j-1} \mathbf{W}^{in}\|_* \\ &\leq \|\mathbf{W}^{out}\|_* \|\mathbf{A}^{j-1}\|_* \|\mathbf{W}^{in}\|_* \\ &\leq \|\mathbf{W}^{out}\|_* \|\mathbf{A}\|_*^{j-1} \|\mathbf{W}^{in}\|_*.\end{aligned}\quad (62)$$

$\|\cdot\|_*$ denotes an induced matrix norm inherited by the corresponding vector norm. That is, allowing a (possibly not square) matrix B mapping between vector spaces V_1, V_2 , by $B : V_1 \rightarrow V_2$, then $\|B\|_{*,V_1,V_2} := \sup_{\|x\|_{*,V_1}=1} \|Bx\|_{*,V_2}$, and $\|\cdot\|_{*,V_i}$ describes the vector norm in V_i . For example, typical favorite vector norms include the Euclidean norm, $\|\cdot\|_2$; the 1-norm, $\|\cdot\|_1$; and the ∞ -norm, $\|\cdot\|_\infty$. The specific vector norm is not important; thus, the noncommittal notation $*$ for good reason is to be described in a moment. For simplicity of notation in Eq. (B12), we omit emphasis of domain V_1 and range V_2 vector spaces of each matrix operator, understanding that dimensionality of the induced vector norms depends on the matrix sizes and ranks.

Inspecting the term, $\|\mathbf{A}\|_*^{j-1}$, in Eq. (62), we can bind by its eigenvalues. For any chosen $\epsilon > 0$,³¹ there exists some induced matrix norm, $\|\cdot\|_*$ (which is why we used the noncommittal notation $*$) such that $\|\mathbf{A}\|_* \leq \rho(\mathbf{A}) + \epsilon$, where $\rho(\mathbf{A}) = \max_{|\lambda_i|} |\lambda_i|$ is the spectral radius defined as the largest magnitude of eigenvalues, noting that \mathbf{A} is square. By theorem,³¹ if $\rho(\mathbf{A}) < 1$, then there exists an induced norm such that in terms of that norm, $\|\mathbf{A}\|_* < 1$, which implies $\|\mathbf{A}^n\|_* \leq \|\mathbf{A}\|_*^n \rightarrow 0$ as $n \rightarrow \infty$. Therefore, by equivalence of norms, $\mathbf{A}^n \rightarrow 0$. This last statement is stronger than a convergence in norm, but rather, it is a componentwise statement of convergence of the matrix to the zero matrix with respect to exponentiation under the condition of the spectral radius bounded by 1.

A result regarding the VAR matrix mentioned in Eqs. (20) and (62) that as long as the readout matrix is bounded $\|\mathbf{W}^{out}\| < C < \infty$ and $\rho(\mathbf{A}) < 1$, the coefficient matrices then become increasingly close to the zero matrix,

$$a_j \rightarrow 0, \text{ as } j \rightarrow \infty, \quad (63)$$

stated strongly as componentwise convergence of the matrices. However, this is not to say that convergence needs to be monotone or that the bound must be sharp. What is interesting in this statement is that the distribution from which the randomly chosen matrix \mathbf{A} is drawn controls the spectral radius $\rho(\mathbf{A})$, which in turn dominates the VAR matrices a_j . This is agreeable with the general VAR theory reviewed in Sec. V that a general VAR(∞) is well approximated by a finite VAR(k).

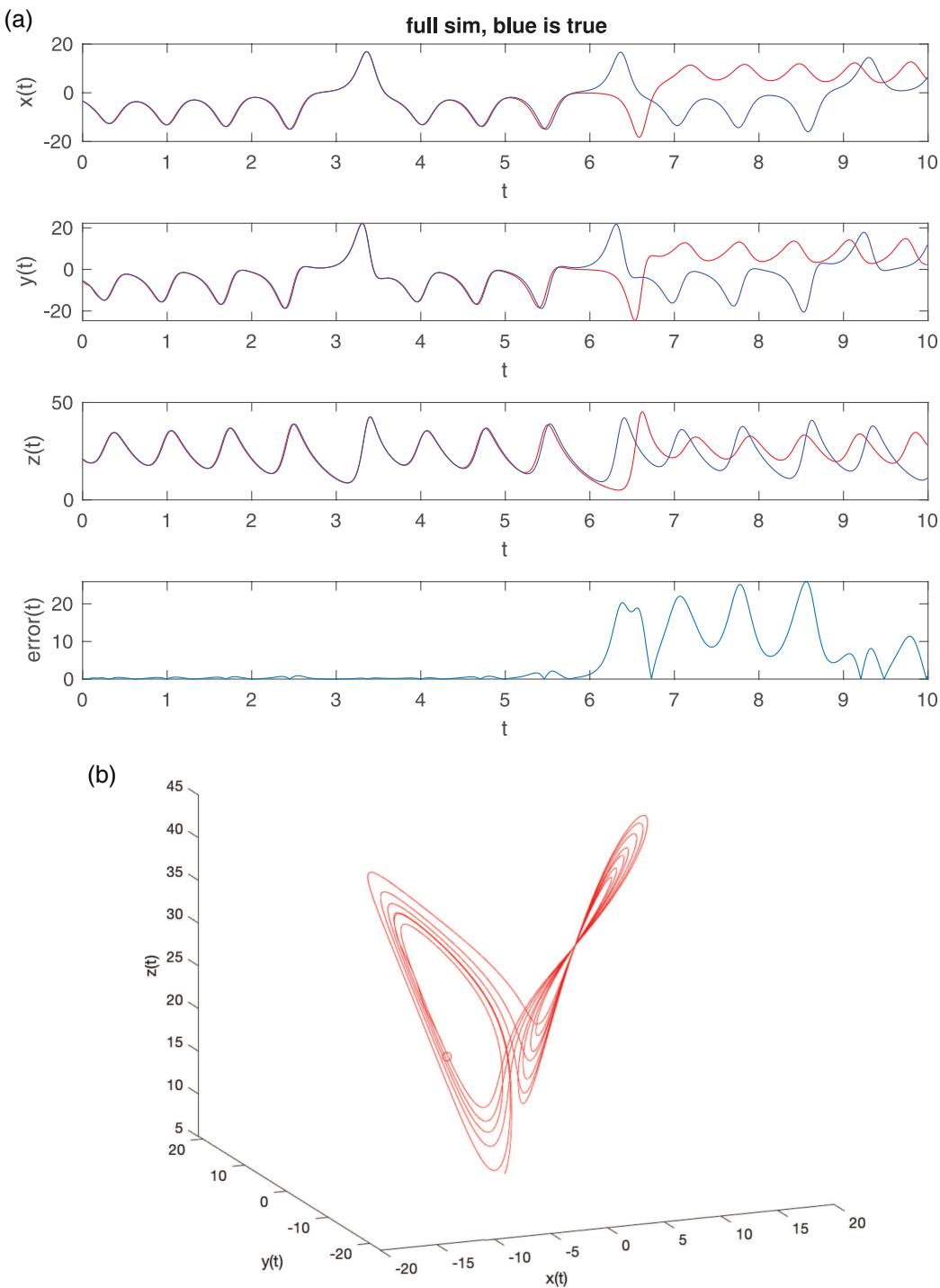


FIG. 6. Lorenz time-series with nonlinear RC, with $q(x) = \tanh(x)$, of size $d_r = 1000$, and quadratic readout, \mathbf{W}^{out} , fitted to input $\mathbf{R} = [\mathbf{R}_1; \mathbf{R}_2]$ from Eq. (B1). The top three time-series are state variables; blue curves show forecasts whereas red shows true data. Bottom, error shown, growing from initial seed. (Right) The phase space presentation of the forecast variables, $[x(t), y(t), z(t)]$. Error performance is excellent. Even once the error has grown, the produced attractor seems true to the forecast Lorenz. Compare to Figs. 7 and 8.

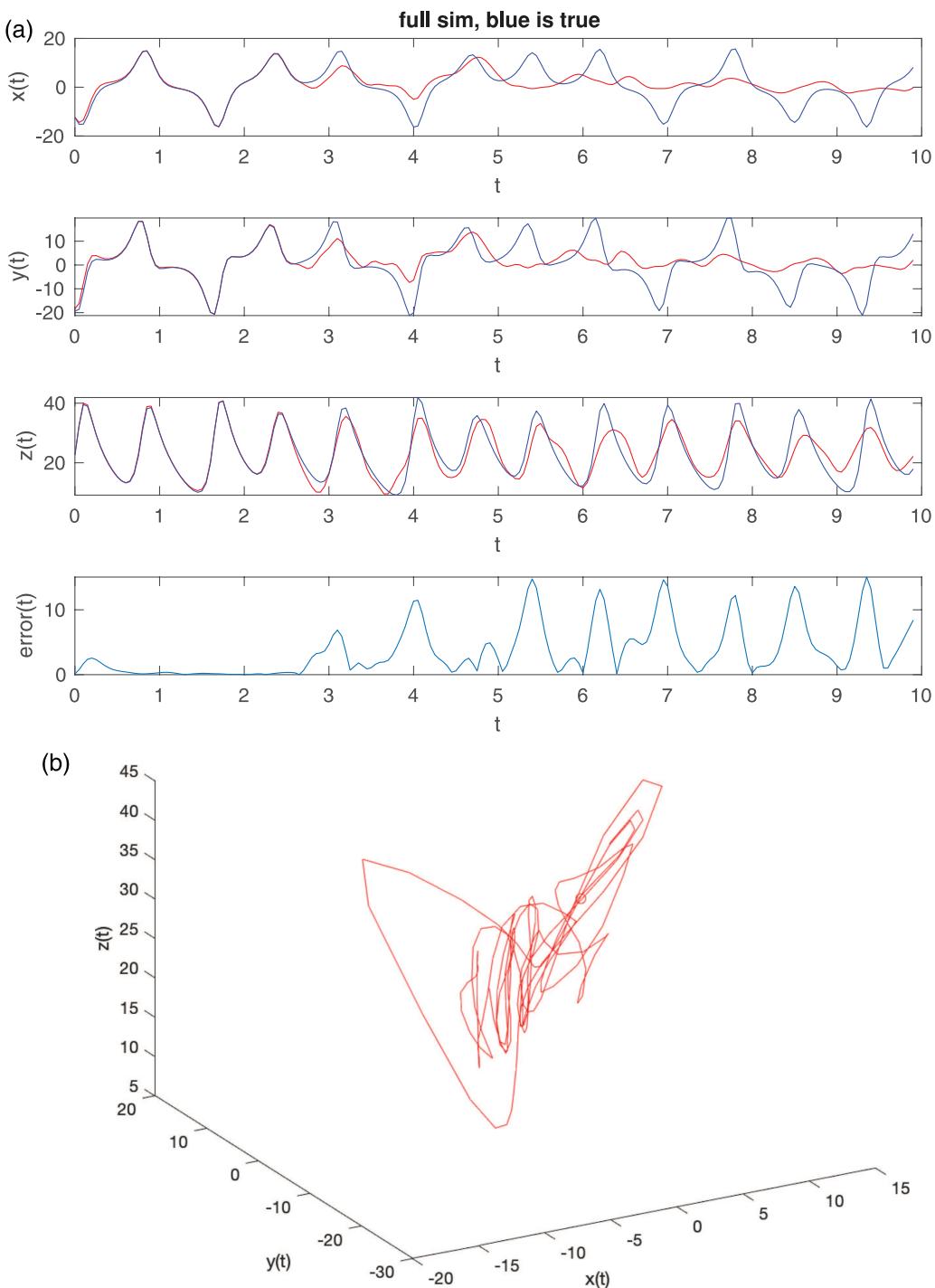


FIG. 7. Lorenz time-series with fully linear RC, so $q(x) = x$ of size $d_r = 1000$, and linear readout. (Left) The top three time-series are state variables; blue curves show forecasts, whereas red shows true data. Bottom, error shown, growing from initial seed. This fully linear RC, being equivalent to a VAR, does produce good forecasts for a finite time even if for a shorter time than that of the nonlinear methods of Figs. 6 and 8. (Right) The phase space presentation of the forecast variables, $[x(t), y(t), z(t)]$. Now, unlike the nonlinear RC or the nonlinear readout cases, despite forecasts for a little while, once errors have occurred in forecasting the time-series of the fully linear RC, the form of the attractor is entirely wrong.

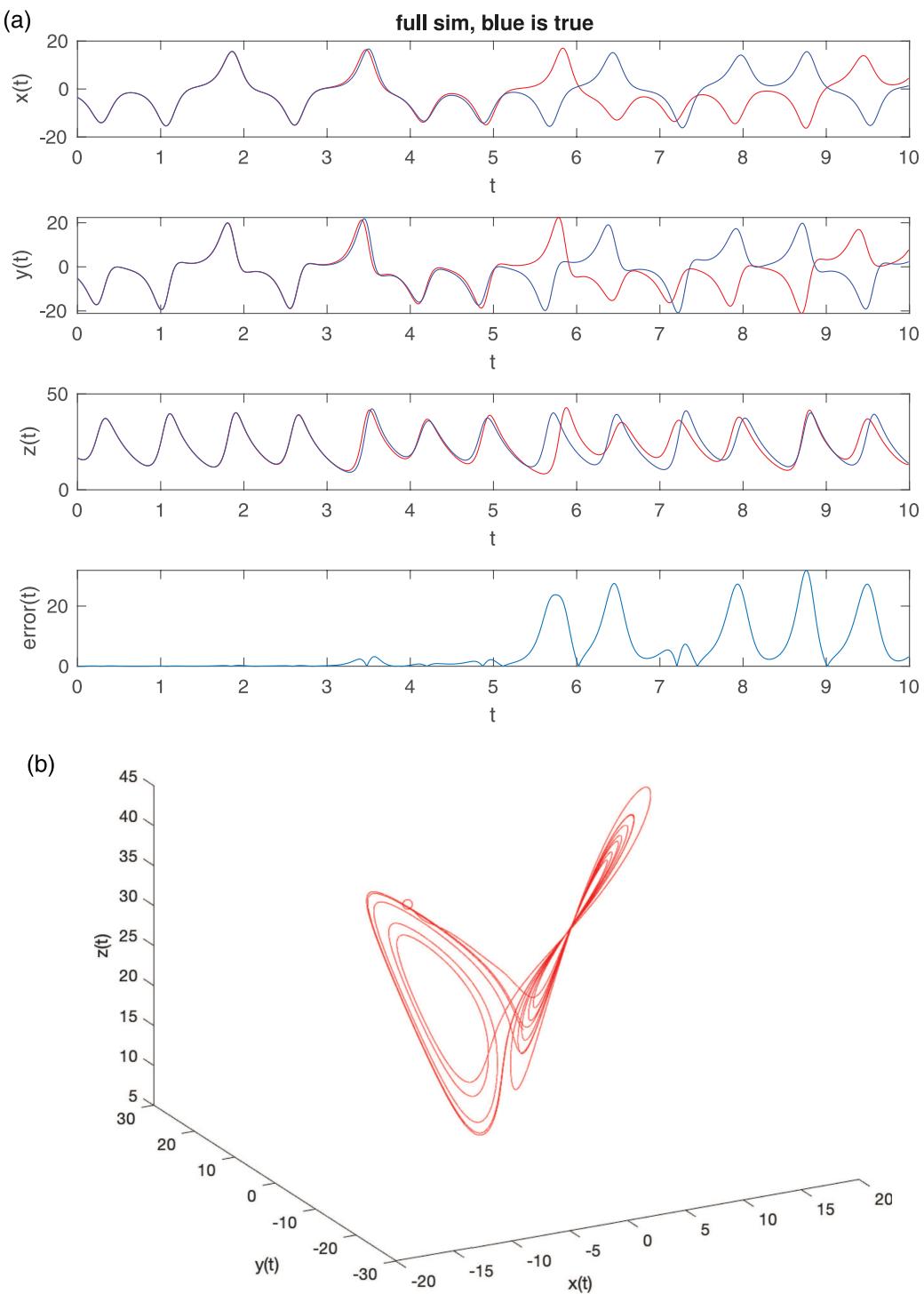


FIG. 8. Lorenz time-series with linear RC, so $q(x) = x$ of size $d_r = 1000$, but quadratic readout, \mathbf{W}^{out} , fitted to input $\mathbf{R} = [\mathbf{R}_1; \mathbf{R}_2]$ from Eq. (B1). (Left) The top three time-series are state variables; blue curves show forecasts, whereas red shows true data. Bottom, error shown, growing from initial seed. (Right) The phase space presentation of the forecast variables, $[x(t), y(t), z(t)]$. As in the full nonlinear RC method of Fig. 6, the attractor seems true even once errors have grown. Compare to Figs. 6 and 7.

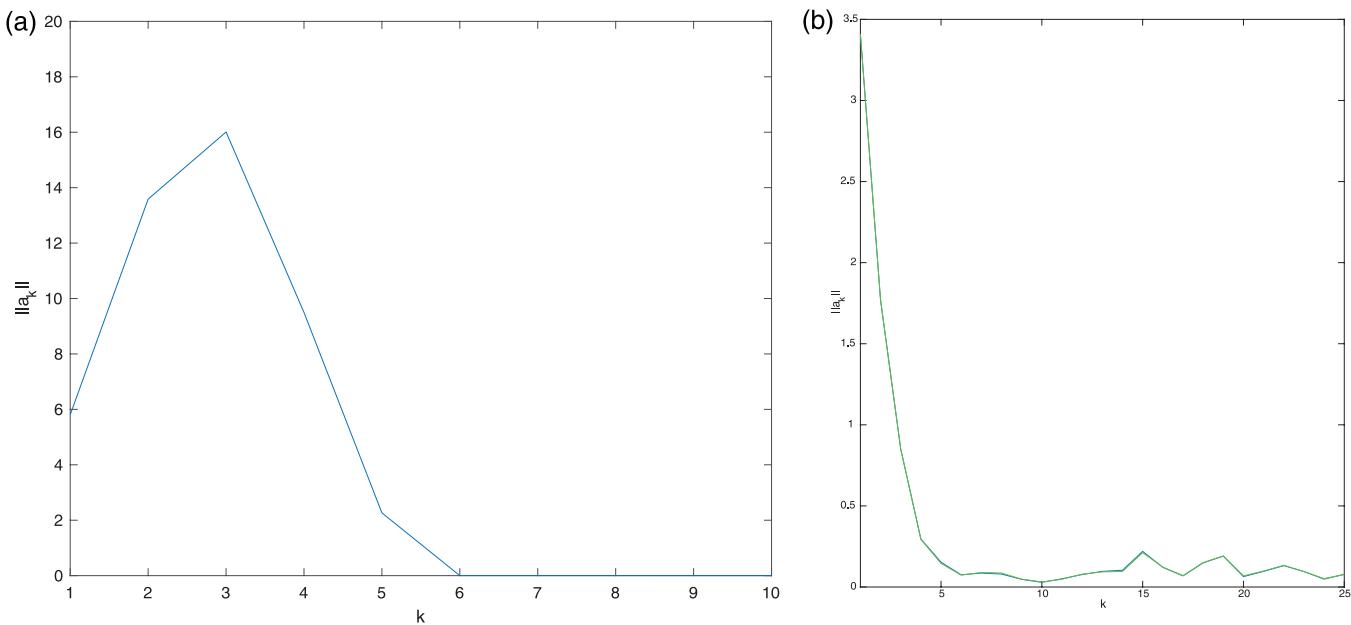


FIG. 9. Fading memory as observed by the magnitude of $\text{VAR}(k)$ matrices a_k , as k vs $\|a_k\|_2$ induced norm, of (top) the Mackey–Glass system, the $\text{VAR}(6)$ model is descriptive since $\|a_6\|_2$ is already relatively small. (Bottom) The Lorenz system where we see that after $k = 9$, coefficient matrices $\|a_k\|_2$ are relatively small, suggesting that the $\text{VAR}(9)$ description is close to the large k model. However, we do not see convergence to zero suggested by Eq. (62), which we attribute to numerical instabilities since the details of the computed a_k seem to vary with details of the orbit data, whereas the head (the early terms) remains stable. Shown in a green curve are the a_k computed by Eq. (62) involving the RC, but the blue curve directly as a VAR fit [Eq. (26)], and we see that these closely agree as the two curves are almost coincident.

B. Experiments on fading memory of the RC

Consider the actual computed a_j matrices and their corresponding induced norms $\|a_j\|_2$ as computed from the example of chaotic systems studied in Sec. VIII, which were the Mackey–Glass system and the Lorenz system.

In Fig. 9 (top), a $(k, \|a_k\|_2)$ curve is shown as computed from Eq. (62) in the case of a RC model derived from the Mackey–Glass equation, corresponding to RC forecasts of the same system in Fig. 1. We see a pronounced fading memory as $\|a_k\|_2$ has diminished to negligible values by $k \geq 6$. This is not unexpected since the attractor of these equations shown in Fig. 1 illustrates primarily a rotation and otherwise weak chaos in the sense that while the embedding dimension may be $d = 4$, the Lyapunov exponent is relatively small even if positive, with $\lambda_{\max} = 0.0058$.⁹⁰

Considering the memory curve $(k, \|a_k\|_2)$ of the Lorenz system in Fig. 9 (bottom), we see that initially, $\|a_k\|_2$ decreases quickly to small values by $k = 9$, but then the computed values fail to decrease further for larger k . We attribute this to difficulties of long range forecasting of this highly chaotic system. Another way of realizing this point is that the details of the tail of the large k a_k values vary with different data samples even though the front part of the series (small k) is stable across samples. Interestingly, nonetheless, as an affirmation of the theoretical discussion, the green curve derived from Eq. (62) involving the RC and the blue curve derived as a VAR fit [Eq. (26)] almost entirely coincide, the

difference being likely due to numerical estimation issues for very large k .

X. CONCLUSION

The success of machine learning and artificial neural networks has led to a clear and overwhelming widely adopted wave across so many areas where data are relevant and patterns are of interest. Dynamical systems are no exception, and forecasting a dynamical system is a specific application that is broadly relevant and of interest to us here. The RNN framework is particularly relevant for dynamical systems since the reserve memory aspect of the concept allows for a good framework and some aspects of delay embedding. However, while the RNN tends to have many parameters to fit, with the danger of overfitting always present, and in any case the large cost of optimization in the training phase, there is a surprising shortcut. The echo-state/reservoir computing concepts presume to choose the weights for the input layer and the internal layer entirely randomly. Then, only the output layer is trained. Furthermore, output layer training will be by linear algebraic manipulations toward a least squares solution rather than the usual nonlinear optimization necessary for the many parameters of the full nonlinear RNN. That this would allow for huge computational savings is clear. What is perhaps a surprise is how this gross simplification still yields useful results. While there have been a number of studies experimentally

describing how to choose better random processes to define the random parameters, e.g., such as to emphasize sparsity or to control the spectral radius and other properties, in this work, we have taken a different approach, which is not specifically to improve the performance of the concept but instead to give a partial explanation as to how the concept can work at all. After all, at first glance, it may seem that it would be impossible that such a simplification could work. In this work, we have simplified the RC concept, allowing for the activation function to be an identity function instead of the more typical sigmoidal function. In this case, the process is entirely linear and so easier to study. As it turns out, the RC still performs reasonably well for short-term forecasts, and it certainly leads to easier analysis. Herein, we prove that the linear RC is in fact directly related to the more matured topic of VAR, vector autoregressive time-series forecasting, and with all the related theory including the Wold theorem as a representation theorem, which, therefore, now applies to the RC. Also, we are able to make a direct connection to the increasingly popular DMD theory. Furthermore, the commonly used fitting upon readout of linear and Hadamard quadratic observations of the reservoir states yields a nonlinear VAR (NVAR) allowing for all quadratic monomial term generalization of the VAR result. This NVAR version apparently not only makes competitive forecasts, but also the errors seem to respect the original attractor in the case of the Lorenz system.

ACKNOWLEDGMENTS

The author has received funding from the Army Research Office (No. N68164-EG) and also DARPA. The author would also like to sincerely thank Daniel J. Gauthier, Aaron Griffith, and Wendson A. S. Barbosa of Ohio State University for their generous and helpful feedback concerning this work.

APPENDIX A: REVIEW OF REGULARIZED PSEUDO-INVERSE

We review how to numerically and stably compute the pseudo-inverse by the singular value decomposition, with regularized singular values (SVD). Reviewing the matrix theory of regularized pseudo-inverses for general matrices, if

$$Xb = z, \quad (\text{A1})$$

$X_{n \times p}, b_{p \times 1}, z_{n \times 1}$ and if the SVD is $X = U\Sigma V$, with orthogonal matrices, $n \times n$, then U satisfies $UU^T = U^TU = I$ and $p \times p$, V satisfies $VV^T = V^TV = I$, and Σ is a $n \times p$ “diagonal” matrix of singular values, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0 \geq \sigma_{r+1} \geq \dots \geq \sigma_p \geq 0$,

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_p \end{bmatrix} \text{ if } n = p, \Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \vdots \end{bmatrix} \text{ if } n > p, \\ \Sigma = \begin{bmatrix} \sigma_1 & & \dots \\ & \ddots & \dots \\ & & \sigma_p \end{bmatrix} \text{ if } n < p. \quad (\text{A2})$$

Then,

$$X^\dagger := (X^T X)^{-1} X^T$$

$$= V \Sigma^\dagger U^T, \text{ where } \Sigma^\dagger = \begin{bmatrix} \frac{1}{\sigma_1} & & & \\ & \ddots & & \\ & & \frac{1}{\sigma_p} & \\ & \vdots & \vdots & \vdots \end{bmatrix}, \text{ in the } n > p \text{ case.} \quad (\text{A3})$$

The least squares estimator of $Xb = z$ is

$$b^* = (X^T X)^{-1} X^T z := X^\dagger z, \quad (\text{A4})$$

and we write the ridge regression Tikhonov regularized solution,

$$b_\lambda^* = (X^T X + \lambda I)^{-1} X^T z = V(\Sigma^T \Sigma + \lambda I)^{-1} \Sigma^T U^T z := X_\lambda^\dagger z. \quad (\text{A5})$$

The regularized pseudo-inverse X_λ^\dagger is better stated in terms of the regularized singular values by

$$\Sigma_\lambda^\dagger := (\Sigma^T \Sigma + \lambda I)^{-1} \Sigma^T \\ = \begin{bmatrix} \frac{\sigma_1}{\sigma_1^2 + \lambda} & & & \\ & \ddots & & \\ & & \frac{\sigma_p}{\sigma_p^2 + \lambda} & \\ & \vdots & \dots & \end{bmatrix} \text{ in the } n > p \text{ case,} \quad (\text{A6})$$

and then,

$$b_\lambda^* = X_\lambda^\dagger z = V \Sigma_\lambda^\dagger U^T z. \quad (\text{A7})$$

Throughout, since we will always refer to regularized pseudo-inverses, we will not emphasize this by abusing notation, allowing that b^* denotes b_λ^* even if only a very small $\lambda > 0$ is chosen, unless otherwise stated, $\lambda = 1.0 \times 10^{-8}$. This mitigates the tendency of overfitting or likewise stated in terms of zero or almost zero singular values that would otherwise appear in the denominators of Σ^\dagger . The theory is similar for $n = p$ and $n < p$, as well as the scenario where z is not just a vector but a matrix, and likewise as in Eq. (9), where we refer to the transpose scenario.

APPENDIX B: ON QUADRATIC NVAR CONNECTION TO RC

In this appendix, we give details claimed in Sec. VI that a linear RC with the Hadamard quadratic nonlinear readout also corresponds to a VAR like entity from stochastic process modeling of time-series. Now, however, a quadratic type nonlinear VAR results, which is an NVAR. This is a generalization of the linear VAR discussion of Sec. IV.

A commonly used scenario of RC⁵⁶ is to fit \mathbf{W}^{out} not just to \mathbf{r} data but also to $\mathbf{r} \circ \mathbf{r}$, where \circ denotes the Hadamard product (implemented in array languages such as Matlab by “array arithmetic” using the “ $*$ ” notation with the dot, in place of what would otherwise be “ $**$ ” for standard matrix multiplication). For a vector $\mathbf{r} = [r_1 | r_2 | \cdots | r_{d_r}]^T$, this is defined as component-wise operations: $\mathbf{r} \circ \mathbf{r} = [r_1^2 | r_2^2 | \cdots | r_{d_r}^2]^T$. The reason for using nonlinear terms is cited as improved performance, allowing for matching the parity of the process. In this case, we rename what before we called \mathbf{R} to now being called \mathbf{R}_1 . Therefore, Eq. (8) is replaced with

$$\begin{aligned}\mathbf{R}_1 &= [\mathbf{r}_k \quad |\mathbf{r}_{k+1}| \cdots |\mathbf{r}_N], \\ \mathbf{R}_2 &= [\mathbf{r}_k \circ \mathbf{r}_k \quad |\mathbf{r}_{k+1} \circ \mathbf{r}_{k+1}| \cdots |\mathbf{r}_N \circ \mathbf{r}_N], \\ \mathbf{R} &= \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix}.\end{aligned}\quad (\text{B1})$$

Then, Eq. (9) remains written as before, $\mathbf{W}^{out} := \mathbf{X}\mathbf{R}^T(\mathbf{R}\mathbf{R}^T + \lambda\mathbf{I})^{-1}$, but now since \mathbf{R} is $2d_r \times N - k$, then \mathbf{W}^{out} is $d_x \times 2d_r$. For convenience, in the rest of this section, partition these matrices \mathbf{W}^{out} into top and bottom half portions. These, as we show, act on linear and quadratic terms of the corresponding NVAR,

$$\mathbf{W}^{out} = \begin{bmatrix} \mathbf{W}_1^{out} \\ \mathbf{W}_2^{out} \end{bmatrix}, \quad (\text{B2})$$

each of size $d_x \times d_r$.

First, note an identity of how the Hadamard product distributes with standard matrix–vector multiplication. Let $\mathbf{w} = [w_1 | w_2 | \cdots | w_n]^T$ be a vector with scalar vector components w_i and B be a general $m \times n$ matrix. Let $B = [\mathbf{b}_1 | \mathbf{b}_2 | \cdots | \mathbf{b}_n]$ be written in terms of the column vectors \mathbf{b}_j of B . Then,

$$\begin{aligned}\mathbf{B}\mathbf{w} \circ \mathbf{B}\mathbf{w} &= (w_1\mathbf{b}_1 + w_2\mathbf{b}_2 | \cdots | w_n\mathbf{b}_n) \circ (w_1\mathbf{b}_1 + w_2\mathbf{b}_2 | \cdots | w_n\mathbf{b}_n) \\ &= [\mathbf{b}_1 \circ \mathbf{b}_1 | \mathbf{b}_2 \circ \mathbf{b}_2 | \cdots | \mathbf{b}_n \circ \mathbf{b}_n] \begin{bmatrix} w_1^2 \\ w_1 w_2 \\ \vdots \\ w_1 w_n \\ w_2 w_1 \\ w_2^2 \\ \vdots \\ w_n^2 \end{bmatrix} := P_2(B, B)\mathbf{p}_2(\mathbf{w}, \mathbf{w}).\end{aligned}\quad (\text{B3})$$

Thus, the Hadamard operator distributes through matrix multiplication to be written purely as matrix multiplication with carefully stated matrices. We have defined the matrix of Hadamard products as $P_2(B, B)$, which is a $m \times n^2$ matrix by the matrix function defined in Eq. (B3),

$$P_2 : \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n^2}, \quad (\text{B4})$$

and vector function also in Eq. (B3),

$$\begin{aligned}p_2(\mathbf{v}, \mathbf{w}) &: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{n^2}, \\ (\mathbf{v}, \mathbf{w}) &\mapsto [v_1 w_1 | v_1 w_2 | \cdots | v_1 w_n | v_2 w_1 | v_2 w_2 | \cdots | v_n w_n]^T,\end{aligned}\quad (\text{B5})$$

to be the $n^2 \times 1$ vector of all quadratic combinations suggested in the equation above, $p_2 : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{n^2}$. By this notation, we will state for convenience identity operators, $P_1(B) = B$ and $p_1(\mathbf{w}) = \mathbf{w}$. Higher order operators follow similarly, but we will not need these here.

With this notation, we can proceed comparably to Eqs. (14)–(18) by tracking iterations of the RC but with quadratic readout and with the terms in the \circ product for use in building \mathbf{W}^{out} to be used in the readout. Let $\mathbf{r}_1 = 0$. Then,

$$\begin{aligned}\mathbf{r}_2 \circ \mathbf{r}_2 &= (\mathbf{W}^{in}\mathbf{x}_1) \circ (\mathbf{W}^{in}\mathbf{x}_1) \\ &= P_2(\mathbf{W}^{in}, \mathbf{W}^{in})\mathbf{p}_2(\mathbf{x}_1), \\ \mathbf{r}_3 \circ \mathbf{r}_3 &= (\mathbf{A}\mathbf{W}^{in}\mathbf{x}_1 + \mathbf{W}^{in}\mathbf{x}_2) \circ (\mathbf{A}\mathbf{W}^{in}\mathbf{x}_1 + \mathbf{W}^{in}\mathbf{x}_2) \\ &= (\mathbf{A}\mathbf{W}^{in}\mathbf{x}_1) \circ (\mathbf{A}\mathbf{W}^{in}\mathbf{x}_1) + (\mathbf{A}\mathbf{W}^{in}\mathbf{x}_1) \circ (\mathbf{W}^{in}\mathbf{x}_2) \\ &\quad + (\mathbf{W}^{in}\mathbf{x}_2) \circ (\mathbf{A}\mathbf{W}^{in}\mathbf{x}_1) + (\mathbf{W}^{in}\mathbf{x}_2) \circ (\mathbf{W}^{in}\mathbf{x}_2) \\ &= P_2(\mathbf{A}\mathbf{W}^{in}, \mathbf{A}\mathbf{W}^{in})\mathbf{p}_2(\mathbf{x}_1, \mathbf{x}_1) + P_2(\mathbf{A}\mathbf{W}^{in}, \mathbf{W}^{in})\mathbf{p}_2(\mathbf{x}_1, \mathbf{x}_2) \\ &\quad + P_2(\mathbf{W}^{in}, \mathbf{A}\mathbf{W}^{in})\mathbf{p}_2(\mathbf{x}_2, \mathbf{x}_1) + P_2(\mathbf{W}^{in}, \mathbf{W}^{in})\mathbf{p}_2(\mathbf{x}_2, \mathbf{x}_2), \\ &\vdots \\ \mathbf{r}_{k+1} \circ \mathbf{r}_{k+1} &= \sum_{i=1}^k (A^{i-1}\mathbf{W}^{in}\mathbf{x}_{k+1-i}) \circ \left(\sum_{j=1}^k A^{j-1}\mathbf{W}^{in}\mathbf{x}_{k+1-j} \right) \\ &= \sum_{i,j=1}^k P_2(A^{i-1}\mathbf{W}^{in}, A^{j-1}\mathbf{W}^{in})\mathbf{p}_2(\mathbf{x}_{k+1-i}, \mathbf{x}_{k+1-j}) \quad (\text{B6}) \\ &:= \mathbb{A}_2[\mathbb{X}_2]_k.\end{aligned}\quad (\text{B7})$$

That is, \mathbb{A} defined in Eq. (28) is $d_r \times kd_x$, and analogously,

$$\begin{aligned}\mathbb{A}_2 &= [P_2(\mathbf{W}^{in}, \mathbf{W}^{in}) | P_2(\mathbf{A}\mathbf{W}^{in}, \mathbf{W}^{in}) | P_2(\mathbf{A}^2\mathbf{W}^{in}, \mathbf{W}^{in}) | \cdots \\ &\quad \cdots | P_2(\mathbf{A}^{k-1}\mathbf{W}^{in}, \mathbf{W}^{in}) | P_2(\mathbf{W}^{in}, \mathbf{A}\mathbf{W}^{in}) | P_2(\mathbf{A}\mathbf{W}^{in}, \mathbf{A}\mathbf{W}^{in}) \\ &\quad \times | P_2(\mathbf{A}^2\mathbf{W}^{in}, \mathbf{A}\mathbf{W}^{in}) | \cdots \\ &\quad \cdots | P_2(\mathbf{A}^{k-2}\mathbf{W}^{in}, \mathbf{A}^{k-1}\mathbf{W}^{in}) | P_2(\mathbf{A}^{k-1}\mathbf{W}^{in}, \mathbf{A}^{k-1}\mathbf{W}^{in})]\end{aligned}\quad (\text{B8})$$

is a $d_r \times kd_x^2$ matrix.

Similarly, where \mathbb{X} is a $(kd_x) \times (N - k)$ matrix of data, \mathbb{X}_2 is a $(kd_x^2) \times (N - k)$ matrix of data, but quadratic forms, analogous to the $(kd_x) \times (N - k)$ array \mathbb{X} from Eqs. (23) and (24), and $[\mathbb{X}_2]_k$ is the k th column,

$$\mathbb{X}_1 = \begin{bmatrix} | & | & \vdots & | \\ \mathbf{x}_k & \mathbf{x}_{k+1} & \cdots & \mathbf{x}_{N-1} \\ | & | & \vdots & | \\ \mathbf{x}_{k-1} & \mathbf{x}_k & \cdots & \mathbf{x}_{N-2} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{N-k} \\ | & | & \vdots & | \end{bmatrix}, \mathbb{X}_2 = \begin{bmatrix} p_2(\mathbf{x}_k, \mathbf{x}_k) & p_2(\mathbf{x}_{k+1}, \mathbf{x}_{k+1}) & \cdots & p_2(\mathbf{x}_{N-1}, \mathbf{x}_{N-1}) \\ | & | & \vdots & | \\ p_2(\mathbf{x}_{k-1}, \mathbf{x}_k) & p_2(\mathbf{x}_k, \mathbf{x}_{k+1}) & \cdots & p_2(\mathbf{x}_{N-2}, \mathbf{x}_{N-1}) \\ \vdots & \vdots & \vdots & \vdots \\ p_2(\mathbf{x}_1, \mathbf{x}_k) & p_2(\mathbf{x}_2, \mathbf{x}_{k+1}) & \cdots & p_2(\mathbf{x}_{N-k-1}, \mathbf{x}_{N-1}) \\ | & | & \vdots & | \\ p_2(\mathbf{x}_k, \mathbf{x}_{k-1}) & p_2(\mathbf{x}_{k+1}, \mathbf{x}_k) & \cdots & p_2(\mathbf{x}_{N-1}, \mathbf{x}_{N-2}) \\ | & | & \vdots & | \\ p_2(\mathbf{x}_{k-1}, \mathbf{x}_{k-1}) & p_2(\mathbf{x}_{k+1}, \mathbf{x}_{k-1}) & \cdots & p_2(\mathbf{x}_{N-2}, \mathbf{x}_{N-2}) \\ \vdots & \vdots & \vdots & \vdots \\ p_2(\mathbf{x}_1, \mathbf{x}_1) & p_2(\mathbf{x}_2, \mathbf{x}_2) & \cdots & p_2(\mathbf{x}_{N-k}, \mathbf{x}_{N-k}) \\ | & | & \vdots & | \end{bmatrix}. \quad (B9)$$

Now, we write $\mathbb{X} = \begin{bmatrix} \mathbb{X}_1 \\ \mathbb{X}_2 \end{bmatrix}$.

With Eqs. (9) and (B1) in mind and with $[\mathbf{R}]_2$, the second column of \mathbf{R} , being $[\mathbf{R}]_2 = [\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}]$, we generalize the VAR stated in Eq. (19). The quadratic NVAR follows:

$$\begin{aligned} \mathbf{y}_{\ell+1} &= \mathbf{W}^{out}[\mathbf{R}]_2 \\ &= \mathbf{W}_1^{out} \sum_{j=1}^{\ell} \mathbf{A}^{j-1} \mathbf{W}^{in} \mathbf{x}_{\ell-j+1} + \mathbf{W}_2^{out} \sum_{i,j=1}^{\ell} P_2(A^{i-1} \mathbf{W}^{in}, A^{j-1} \mathbf{W}^{in}) \\ &\quad \times p_2(\mathbf{x}_{\ell+1-i}, \mathbf{x}_{\ell+1-j}) \\ &= \mathbf{W}_1^{out} \mathbf{A}^{\ell-1} \mathbf{W}^{in} \mathbf{x}_1 + \mathbf{W}_1^{out} \mathbf{A}^{\ell-2} \mathbf{W}^{in} \mathbf{x}_2 + \cdots + \mathbf{W}_1^{out} \mathbf{A} \mathbf{W}_1^{in} \mathbf{x}_{\ell-1} \\ &\quad + \mathbf{W}_1^{out} \mathbf{W}^{in} \mathbf{x}_{\ell} + \sum_{i,j=1}^{\ell} \mathbf{W}_2^{out} P_2(A^{i-1} \mathbf{W}^{in}, A^{j-1} \mathbf{W}^{in}) \\ &\quad \times p_2(\mathbf{x}_{\ell+1-i}, \mathbf{x}_{\ell+1-j}) \\ &= a_{\ell} \mathbf{x}_1 + a_{\ell-1} \mathbf{x}_2 + \cdots + a_2 \mathbf{x}_{\ell-1} + a_1 \mathbf{x}_{\ell} + a_{2,(\ell,\ell)} p_2(\mathbf{x}_1, \mathbf{x}_1) \\ &\quad + a_{2,(\ell-1,\ell)} p_2(\mathbf{x}_2, \mathbf{x}_1) + \cdots + a_{2,(1,1)} p_2(\mathbf{x}_{\ell}, \mathbf{x}_1), \end{aligned} \quad (B10)$$

with the notation for the ℓ linear coefficient $d_x \times d_x$ matrices as before,

$$a_j = \mathbf{W}_1^{out} \mathbf{A}^{j-1} \mathbf{W}^{in}, j = 1, 2, \dots, \ell, \quad (B11)$$

and now, we have ℓ^2 quadratic term coefficient $d_x \times d_x^2$ matrices,

$$a_{2,(i,j)} = \mathbf{W}_2^{out} P_2(A^{i-1} \mathbf{W}^{in}, A^{j-1} \mathbf{W}^{in}), i, j = 1, \dots, \ell. \quad (B12)$$

This is the generalization of the VAR equation coefficients written explicitly in Eq. (20) to these coefficient matrices of a quadratic NVAR that results in a linear RC with the Hadamard quadratic readout.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- ¹G. Amisano and C. Giannini, *Topics in Structural VAR Econometrics* (Springer Science & Business Media, 2012).
- ²P. Antonik, M. Gulina, J. Pauwels, and S. Massar, “Using a reservoir computer to learn chaotic attractors, with applications to chaos synchronization and cryptography,” *Phys. Rev. E* **98**(1), 012215 (2018).
- ³H. Arbabi and I. Mezic, “Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator,” *SIAM J. Appl. Dyn. Syst.* **16**(4), 2096–2126 (2017).
- ⁴W. E. Arnoldi, “The principle of minimized iterations in the solution of the matrix eigenvalue problem,” *Q. Appl. Math.* **9**(1), 17–29 (1951).
- ⁵C. A. L. Bailer-Jones, D. J. C. MacKay, and P. J. Withers, “A recurrent neural network for modelling dynamical systems,” *Netw. Comput. Neural Syst.* **9**(4), 531–547 (1998).
- ⁶T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis, and P. S. Dokopoulos, “Long-term wind speed and power forecasting using local recurrent neural network models,” *IEEE Trans. Energy Convers.* **21**(1), 273–284 (2006).
- ⁷E. Boltt, “Regularized kernel machine learning for data driven forecasting of chaos,” *Annual Review of Chaos Theory, Bifurcations and Dynamical Systems* **9**, 1–26 (2020).

- ⁸E. Boltt, "Geometric considerations of a good dictionary for Koopman analysis of dynamical systems," [arXiv:1912.09570](https://arxiv.org/abs/1912.09570) (2019).
- ⁹E. M. Boltt, "Model selection, confidence and scaling in predicting chaotic time-series," *Int. J. Bifurcation Chaos* **10**(06), 1407–1422 (2000).
- ¹⁰E. M. Boltt, L. Billings, and I. B. Schwartz, "A manifold independent approach to understanding transport in stochastic dynamical systems," *Physica D* **173**(3–4), 153–177 (2002).
- ¹¹E. M. Boltt, Q. Li, F. Dietrich, and I. Kevrekidis, "On matching, and even rectifying, dynamical systems through Koopman operator eigenfunctions," *SIAM J. Appl. Dyn. Syst.* **17**(2), 1925–1960 (2018).
- ¹²E. M. Boltt and N. Santitissadeekorn, *Applied and Computational Measurable Dynamics* (SIAM, 2013).
- ¹³G. E. P. Box, , *Time Series Analysis: Forecasting and Control*, Holden-Day Series in Time Series Analysis, edited by G. E. P. Box, G. M. Jenkins, and G. C. Reinsel (Holden-Day, San Francisco, CA, 1994), pp. 199–201.
- ¹⁴S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with Volterra series," *IEEE Trans. Circuits Syst.* **32**(11), 1150–1161 (1985).
- ¹⁵S. L. Brunton and J. Nathan Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (Cambridge University Press, 2019).
- ¹⁶M. Budinić, R. Mohr, and I. Mezić, "Applied Koopmanism," *Chaos* **22**(4), 047510 (2012).
- ¹⁷M. Buehner and P. Young, "A tighter bound for the echo state property," *IEEE Trans. Neural Netw.* **17**(3), 820–824 (2006).
- ¹⁸D. Canaday, A. Griffith, and D. J. Gauthier, "Rapid time series prediction with a hardware-based reservoir computer," *Chaos* **28**(12), 123119 (2018).
- ¹⁹T. L. Carroll and L. M. Pecora, "Network structure effects in reservoir computers," *Chaos* **29**(8), 083130 (2019).
- ²⁰A. Chattopadhyay, P. Hassanzadeh, D. Subramanian, and K. Palem, "Data-driven prediction of a multi-scale Lorenz 96 chaotic system using a hierarchy of deep learning methods: Reservoir computing, ANN, and RNN-LSTM," [arxiv:1906.08829](https://arxiv.org/abs/1906.08829) (2019).
- ²¹J.-F. Chen, W.-M. Wang, and C.-M. Huang, "Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting," *Electr. Power Syst. Res.* **34**(3), 187–196 (1995).
- ²²E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Using recurrent neural network models for early detection of heart failure onset," *J. Am. Med. Inform. Assoc.* **24**(2), 361–370 (2017).
- ²³J. Connor, L. E. Atlas, and D. R. Martin, "Recurrent networks and NARMA modeling," in *Proceedings of the 4th International Conference on Neural Information Processing Systems* (Advances in Neural Information Processing Systems, 1992), pp. 301–308.
- ²⁴D. Darmon, C. J. Cellucci, and P. E. Rapp, "Information dynamics with confidence: Using reservoir computing to construct confidence intervals for information-dynamical measures," *Chaos* **29**(8), 083113 (2019).
- ²⁵P. De Wilde, *Neural Network Models: An Analysis* (Springer, 1996).
- ²⁶J. Doyne Farmer, "Chaotic attractors of an infinite-dimensional dynamical system," *Physica D* **4**(3), 366–393 (1982).
- ²⁷M. Farzad, H. Tahersima, and H. Khaloozadeh, "Predicting the Mackey glass chaotic time series using genetic algorithm," in *2006 SICE-ICASE International Joint Conference* (IEEE, 2006), pp. 5460–5463.
- ²⁸K.-I. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural Netw.* **6**(6), 801–806 (1993).
- ²⁹C. Galicchio, "Chasing the echo state property," [arXiv:1811.10892](https://arxiv.org/abs/1811.10892) (2018).
- ³⁰D. J. Gauthier, "Reservoir computing: Harnessing a universal dynamical system," *Phys. Rev. Lett.* **120**(2018), 024102 (2018).
- ³¹G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. (Johns Hopkins, 2013).
- ³²L. Gonon and J.-P. Ortega, "Reservoir computing universality with stochastic inputs," *IEEE Trans. Neural Netw. Learn. Syst.* **31**(1), 100–112 (2019).
- ³³L. Gonon and J.-P. Ortega, "Fading memory echo state networks are universal," [arXiv:2010.12047](https://arxiv.org/abs/2010.12047) (2020).
- ³⁴A. Griffith, A. Pomerance, and D. J. Gauthier, "Forecasting chaotic systems with very low connectivity reservoir computers," *Chaos* **29**(12), 123108 (2019).
- ³⁵B. J. Grzyb, E. Chinellato, G. M. Wojcik, and W. A. Kaminski, "Which model to use for the liquid state machine?," in *2009 International Joint Conference on Neural Networks* (IEEE, 2009), pp. 1018–1024.
- ³⁶M. Han, Z.-W. Shi, and W. Guo, "Reservoir neural state reconstruction and chaotic time series prediction," *Acta Phys. Sin.* **56**(1), 43–50 (2007).
- ³⁷L. Harrison, W. D. Penny, and K. Friston, "Multivariate autoregressive modeling of fMRI time series," *Neuroimage* **19**(4), 1477–1491 (2003).
- ³⁸A. Hart, J. Hook, and J. Dawes, "Embedding and approximation theorems for echo state networks," *Neural Netw.* **128**, 234–247 (2020).
- ³⁹A. G. Hart, J. L. Hook, and J. H. P. Dawes, "Echo state networks trained by Tikhonov least squares are $l_2(\mu)$ approximators of ergodic dynamical systems," [arXiv:2005.06967](https://arxiv.org/abs/2005.06967) (2020).
- ⁴⁰D. Hartman and L. K. Mestha, "A deep learning framework for model reduction of dynamical systems," in *2017 IEEE Conference on Control Technology and Applications (CCTA)* (IEEE, 2017), pp. 1917–1922.
- ⁴¹S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.* **9**(8), 1735–1780 (1997).
- ⁴²J.-Q. Huang and F. L. Lewis, "Neural-network predictive control for nonlinear dynamic systems with time-delay," *IEEE Trans. Neural Netw.* **14**(2), 377–389 (2003).
- ⁴³H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," German National Research Centre for Information Technology, GMD Technical Reports, Bonn, Germany, 2001, Vol. 148, p. 13.
- ⁴⁴H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science* **304**(5667), 78–80 (2004).
- ⁴⁵J. Jiang and Y.-C. Lai, "Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius," *Phys. Rev. Res.* **1**(3), 033056 (2019).
- ⁴⁶M. B. Kennel and S. Isabelle, "Method to distinguish possible chaos from colored noise and to determine embedding parameters," *Phys. Rev. A* **46**(6), 3111 (1992).
- ⁴⁷M. Kimura and R. Nakano, "Learning dynamical systems by recurrent neural networks from orbits," *Neural Netw.* **11**(9), 1589–1599 (1998).
- ⁴⁸S. N.. Kumpati, P. Kannan *et al.*, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.* **1**(1), 4–27 (1990).
- ⁴⁹J. Nathan Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems* (SIAM, 2016).
- ⁵⁰M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognit. Lett.* **42**, 11–24 (2014).
- ⁵¹A. Lasota and M. C. Mackey, *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics* (Springer Science & Business Media, 2013), Vol. 97.
- ⁵²D. S. Levine, *Introduction to Neural and Cognitive Modeling* (Routledge, 2018).
- ⁵³Q. Li, F. Dietrich, E. M. Boltt, and I. G. Kevrekidis, "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator," *Chaos* **27**(10), 103111 (2017).
- ⁵⁴A. Lichtenberg, "J and Lieberman Ma 1983 regular and stochastic motion," *Appl. Math. Sci.* **38**, 85 (1983).
- ⁵⁵E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmos. Sci.* **20**(2), 130–141 (1963).
- ⁵⁶Z. Lu, B. R. Hunt, and E. Ott, "Attractor reconstruction by machine learning," *Chaos* **28**(6), 061104 (2018).
- ⁵⁷M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural Networks: Tricks of the Trade* (Springer, 2012), pp. 659–686.
- ⁵⁸M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.* **3**(3), 127–149 (2009).
- ⁵⁹H. Lütkepohl, *New Introduction to Multiple Time Series Analysis* (Springer Science & Business Media, 2005).
- ⁶⁰W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.* **14**(11), 2531–2560 (2002).
- ⁶¹M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science* **197**(4300), 287–289 (1977).
- ⁶²S. Marsland, *Machine Learning: An Algorithmic Perspective* (CRC Press, 2015).

- ⁶³D. Michie, D. J. Spiegelhalter, C. C. Taylor *et al.*, "Machine learning," *Neural Stat. Classification* **13**(1994), 1–298 (1994).
- ⁶⁴M. R. Muldoon, D. S. Broomhead, J. P. Huke, and R. Hegger, "Delay embedding in the presence of dynamical noise," *Dyn. Stab. Syst.* **13**(2), 175–186 (1998).
- ⁶⁵M. M. Nelson and W. T. Illingworth, "A practical guide to neural nets," 1991; available at <https://www.osti.gov/biblio/5633084>.
- ⁶⁶S. Ortín González, M. C. Soriano, L. Pesquera González, D. Brunner, D. San Martín Segura, I. Fischer, C. Mirasso, J. M. Gutiérrez Llorente *et al.*, "A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron," *Sci. Rep.* **5**, 14945 (2015).
- ⁶⁷N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," *Phys. Rev. Lett.* **45**(9), 712 (1980).
- ⁶⁸S. Madhavrao Pandit, S.-M. Wu *et al.*, *Time Series and System Analysis with Applications* (Wiley, New York, 1983), Vol. 3.
- ⁶⁹R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning* (PMLR, 2013), pp. 1310–1318.
- ⁷⁰J. 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**(2), 024102 (2018).
- ⁷¹J. Paulsen and D. Tjøstheim, "On the estimation of residual variance and order in autoregressive time series," *J. R. Stat. Soc. Ser. B* **47**(2), 216–228 (1985).
- ⁷²D. Pena, G. C. Tiao, and R. S. Tsay, *A Course in Time Series Analysis* (John Wiley & Sons, 2011), Vol. 322.
- ⁷³D. Qin, "Rise of VAR modelling approach," *J. Econ. Surv.* **25**(1), 156–174 (2011).
- ⁷⁴A. Mohiuddin Rather, A. Agarwal, and V. N. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Syst. Appl.* **42**(6), 3234–3241 (2015).
- ⁷⁵O. A. Rosso, H. A. Larrondo, M. T. Martin, A. Plastino, and M. A. Fuentes, "Distinguishing noise from chaos," *Phys. Rev. Lett.* **99**(15), 154102 (2007).
- ⁷⁶C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, D. Henningson *et al.*, "Spectral analysis of nonlinear flows," *J. Fluid Mech.* **641**(1), 115–127 (2009).
- ⁷⁷T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," *J. Stat. Phys.* **65**(3–4), 579–616 (1991).
- ⁷⁸P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *J. Fluid Mech.* **656**, 5–28 (2010).
- ⁷⁹A. Scott, *Encyclopedia of Nonlinear Science* (Routledge, 2006).
- ⁸⁰C. Serio, "Autoregressive representation of time series as a tool to diagnose the presence of chaos," *Europhys. Lett.* **27**(2), 103 (1994).
- ⁸¹Q. Song and Z. Feng, "Effects of connectivity structure of complex echo state network on its prediction performance for nonlinear time series," *Neurocomputing* **73**(10–12), 2177–2185 (2010).
- ⁸²J. Sun, D. Taylor, and E. M. Bollt, "Causal network inference by optimal causation entropy," *SIAM J. Appl. Dyn. Syst.* **14**(1), 73–106 (2015).
- ⁸³F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980* (Springer, 1981), pp. 366–381.
- ⁸⁴G. C. Tiao and R. S. Tsay, "Consistency properties of least squares estimates of autoregressive parameters in ARMA models," *Ann. Stat.* **11**, 856–871 (1983).
- ⁸⁵H. A. Van der Vorst, *Iterative Krylov Methods for Large Linear Systems* (Cambridge University Press, 2003), Vol. 13.
- ⁸⁶K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nat. Commun.* **5**(1), 1–6 (2014).
- ⁸⁷D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Netw.* **20**(3), 391–403 (2007).
- ⁸⁸P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks," *Proc. R. Soc. A* **474**(2213), 20170844 (2018).
- ⁸⁹P. R. Vlachas, J. Pathak, B. R. Hunt, T. P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos, "Forecasting of spatio-temporal chaotic dynamics with recurrent neural networks: A comparative study of reservoir computing and backpropagation algorithms," [arXiv:1910.05266](https://arxiv.org/abs/1910.05266) (2019).
- ⁹⁰H. Wernecke, B. Sándor, and C. Gros, "Chaos in time delay systems, an educational review," *Phys. Rep.* **824**, 1–40 (2019).
- ⁹¹M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," *J. Nonlinear Sci.* **25**(6), 1307–1346 (2015).
- ⁹²H. O. Andreas Wold, *A Study in the Analysis of Stationary Time Series: With an Appendix* (Almqvist & Wiksell, 1954).
- ⁹³K. Yeo, "Model-free prediction of noisy chaotic time series by deep learning," [arXiv:1710.01693](https://arxiv.org/abs/1710.01693) (2017).
- ⁹⁴K. Yeo and I. Melnyk, "Deep learning algorithm for data-driven simulation of noisy dynamical system," *J. Comput. Phys.* **376**, 1212–1231 (2019).
- ⁹⁵K. Yonemoto and T. Yanagawa, "Estimating the embedding dimension and delay time of chaotic time series by an autoregressive model," *Bull. Inf. Cybern.* **33**(1–2), 53–62 (2001).
- ⁹⁶R. S. Zimmermann and U. Parlitz, "Observing spatio-temporal dynamics of excitable media using reservoir computing," *Chaos* **28**(4), 043118 (2018).