Chaos
An Interdisciplinary Journal of Nonlinear Science

RESEARCH ARTICLE | AUGUST 30 2024

Model reduction of dynamical systems with a novel data-driven approach: The RC-HAVOK algorithm

Special Collection: Data-Driven Models and Analysis of Complex Systems

G. Yılmaz Bingöl ; O. A. Soysal ; E. Günay

Check for updates

Chaos 34, 083143 (2024)
https://doi.org/10.1063/5.0207907

View Online
Export Citation

02 September 2024 10:14:31

Chaos
Special Topic:
Anomalous Diffusion and Fluctuations in Complex Systems and Networks
Submit Today
AIP Publishing

AIP Publishing

Chaos ARTICLE pubs.aip.org/aip/cha

Model reduction of dynamical systems with a novel data-driven approach: The RC-HAVOK algorithm

Cite as: Chaos 34, 083143 (2024); doi: 10.1063/5.0207907
Submitted: 12 March 2024 · Accepted: 7 August 2024 ·
Published Online: 30 August 2024

View Online Export Citation CrossMark

G. Yılmaz Bingöl,[a] O. A. Soysal, and E. Günay

AFFILIATIONS

Department of Electrical and Electronics Engineering, Erciyes University, Kayseri 38039, Türkiye

Note: This paper is part of the Focus Issue: Data-Driven Models and Analysis of Complex Systems.
[a]**Author to whom correspondence should be addressed:** gulnur.yilmaz@erciyes.edu.tr

ABSTRACT

This paper introduces a novel data-driven approximation method for the Koopman operator, called the RC-HAVOK algorithm. The RC-HAVOK algorithm combines Reservoir Computing (RC) and the Hankel Alternative View of Koopman (HAVOK) to reduce the size of the linear Koopman operator with a lower error rate. The accuracy and feasibility of the RC-HAVOK algorithm are assessed on Lorenz-like systems and dynamical systems with various nonlinearities, including the quadratic and cubic nonlinearities, hyperbolic tangent function, and piece-wise linear function. Implementation results reveal that the proposed model outperforms a range of other data-driven model identification algorithms, particularly when applied to commonly used Lorenz time series data.

Published under an exclusive license by AIP Publishing. https://doi.org/10.1063/5.0207907

02 September 2024 10:14:31

Data-driven techniques play a crucial role in scientific research where understanding patterns and making predictions from data are essential tasks. These algorithms are specifically designed to analyze and extract patterns directly from data without relying on explicit models or predefined equations. This scenario is particularly suited for systems where the structure within the data is complex, and traditional analytical approaches may be challenging or impractical. The investigation focuses on the challenges associated with the use of the Koopman operator and approximation techniques, resulting in the introduction of the RC-HAVOK algorithm, which combines Reservoir Computing and the Hankel Alternative View of Koopman. Specific aspects covered in this paper include implementing the HAVOK algorithm to identify finite-dimensional Koopman operator representation, utilizing reservoir computers for dimension reduction, and achieving improvements in approximation error rates and modeling time for certain cases through the proposed RC-HAVOK approach.

I. INTRODUCTION

The study of dynamical systems has been a fascinating area of research in mathematics, physics, and other scientific disciplines.[1–3] Chaotic systems, as components of dynamical systems, demonstrate non-periodic flow, resulting in more complex and unpredictable behavior over time. This feature constitutes one of the main characteristics of chaotic systems, along with a strong sensitivity to initial conditions and a deterministic structure.[4] The high sensitivity to initial conditions in chaotic systems implies that even small variations in the initial state of the system can result in significant deviations in its future behavior.[5] Consequently, analyzing and controlling chaotic systems is a challenging problem, as small uncertainties lead to different outcomes. As a result, recent developments have provided new perspectives to extract meaningful information from observed time series data and approximate system behavior.

The Koopman operator, originally introduced in the 1930s,[6] is another approach that provides a powerful mathematical tool for representing and analyzing nonlinear dynamical systems in a linear framework.[7] The Koopman operator acts on functions defined on the state space of a dynamical system and evolves them forward in time. In an infinite-dimensional setting, this operator represents an infinite number of observables in the state space.[8] The numerical construction of the Koopman operator presents a trade-off, where it may rely on fewer initial conditions but demands longer run times, making it particularly suitable for applications involving physical experiments.[9] This understanding has applications in diverse fields,

Published under an exclusive license by AIP Publishing

including meteorology, mechanics, electronics (including sensors, autonomous vehicles, and high-speed trains), and biology, where nonlinear behavior is prevalent.[10–20]

It is worth noting that Koopman theory includes a linear, yet infinite-dimensional operator that can capture the entire nonlinear dynamics underlying a system.[21] While the Koopman operator itself is infinite-dimensional in theory, approximation methods, particularly data-driven approaches, are often employed to represent the Koopman operator in a finite-dimensional space.[22] This can be achieved through methods such as Dynamic Mode Decomposition (DMD) or Extended Dynamic Mode Decomposition (EDMD).[12,13] These methods provide a finite-dimensional approximation of the Koopman operator, making it computationally feasible to analyze and understand the dynamics of complex systems.

This paper introduces a novel data-driven methodology, RC-HAVOK, with the aim of deriving linear representations of nonlinear dynamical systems. The main point of this algorithm involves merging the reservoir computing method with the traditional HAVOK approach. It is worth noting that a recent study has highlighted a correlation between the Koopman operator and reservoir computing, specifically in scenarios where the reservoir activation function is linear.[23] Conversely, Gulina and Mauroy conducted an investigation into the combination of Extended Dynamic Mode Decomposition and reservoir computing to approximate the Koopman operator.[24] They proposed two methodologies for this purpose. In their study, they noted that the first proposed method may result in a Koopman matrix size that is excessively large. Additionally, they suggested a second method aimed at achieving a low-dimensional approximation, though it comes with increased computational complexity and potentially higher error rates. These prior investigations prompted our exploration of potential connections between the reservoir computing method and HAVOK analysis, which serves as a Koopman operator approximation technique. The primary objective of the RC-HAVOK approach is to derive a reduced-dimensional linear representation of the Koopman operator, aiming for minimized error rates and computational times in specific cases.

The rest of the paper is organized as follows: Sec. II provides background information on approximation methods to the Koopman operator available in the literature. In Sec. III, the motivation behind this study is outlined. Section IV encompasses a general description of the Koopman operator, reservoir computers, and the HAVOK model. Section V details the methodology, introducing the RC-HAVOK algorithm. Implementation results for various dynamical systems are presented in Sec. VI. Finally, Sec. VII offers a discussion, and Sec. VIII provides the conclusion of the paper.

II. A BRIEF REVIEW OF THE APPROXIMATION METHODS TO THE KOOPMAN OPERATOR

One traditional strategy to approximate the Koopman operator is Principal Component Analysis (PCA). PCA aims to simplify high-dimensional data complexity by generating orthogonal modes that efficiently capture the vector energy within a provided dataset. However, the accuracy of reduced-order models derived from PCA may be compromised, and the process of selecting which PCA modes to retain in a reduced-order model is intricate and potentially challenging.[25]

Another widely used data-driven method to find a finite-dimensional representation of the Koopman operator is Dynamic Mode Decomposition. This approach is often applied to empirical data generated by nonlinear dynamical systems and DMD computes the Koopman eigenvalues and modes directly from the given dataset.[26] The eigenvalues indicate the growth or decay rates of the modes, and the eigenvectors represent the spatial structures. Despite the linear assumption, DMD can often provide valuable insights into the dominant linearized features of the nonlinear system. DMD has several variations and extensions each designed to address specific challenges or improve its performance in certain contexts.[15,17,27]

EDMD is an extension of DMD, which computes eigenvalues, eigenfunctions, and modes of the Koopman operator.[28] It involves lifting the data into a higher-dimensional space using nonlinear feature maps before applying DMD. Although this allows for capturing more complex dynamics, EDMD still suffers defining observable particularly when dealing with high-dimensional nonlinear systems.[27] To improve the performance of EDMD, Li and Jiang have developed an iterative approximation algorithm that couples the EDMD and Artificial Neural Network.[29]

SINDy algorithm, which stands for Sparse Identification of Nonlinear Dynamics, employs sparse regression to identify governing equations from data.[30] This technique finds extensive application in identifying linear, nonlinear, and chaotic oscillators, as well as fluid flow.[31,32] Because of the challenges encountered with the conventional SINDy structure, researchers have pursued the development of extensions to the algorithm, aiming to improve its robustness and address any limitations. Kaheman and co-workers introduced the SINDy-PI model to address issues associated with noise sensitivity identified in prior extensions. This model offers a more robust structure, allowing for the identification of implicit dynamics and rational nonlinearities.[33]

The development of the Hankel Alternative View of Koopman method is significant, as it is a data-driven algorithm that draws inspiration from the application of SINDy.[34] HAVOK introduces the use of Hankel matrices, which are structured matrices derived from time series data, to enhance the identification of dynamical systems' underlying equations. This extension allows for improved performance, especially in scenarios where other extensions of the SINDy might face challenges, such as dealing with incomplete or noisy data.[35]

III. MOTIVATION

Overall, the study of dynamical systems continues to be a rich and evolving area of research, with ongoing efforts to deepen our understanding of their dynamics and develop new methods for approximations within the constraints of their inherent unpredictability. Hence, studies based on the Koopman operator and approximation techniques, including neural networks and machine learning approaches, have been examined to identify prevalent challenges. Motivated by this examination, a novel approach, based on the reservoir computers and Hankel Alternative View of Koopman, is introduced. This study specifically addresses the following aspects:

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

- The HAVOK approximation has been utilized for dimension reduction in a dynamical system represented by the Koopman operator. The model's performance has been elevated through the incorporation of reservoir computers.
- The RC-HAVOK algorithm is applied to dynamical systems featuring both smooth and non-smooth activation functions, including quadratic, cubic, hyperbolic tangent, and PWL forms. The feasibility and effectiveness of the proposed model are discussed based on the implementation results.
- In addition to deriving reduced-order models through the proposed algorithm, notable improvements have been achieved in both approximation error rates and modeling time, showcasing enhanced performance in certain aspects.

IV. PRELIMINARIES

A. Koopman operator formalism

The Koopman operator theory describes the linear representation of nonlinear dynamical systems by establishing an infinite-dimensional linear operator. This so-called Koopman operator was introduced in 1931 by B. O. Koopman and has been a focal point of research due to its ability to express nonlinear dynamics within a linear framework.

Consider an autonomous dynamical system given by

$$\dot{x} = F(x(t)), x \in X, \tag{1}$$

where F is a continuously differentiable nonlinear function, $F : X \to X$ represents a nonlinear map, and X is the state space.

Let $g(t)$ is a measurement function of the state x according to dynamics in Eq. (1). Then, the Koopman formalism

$$Kg \triangleq g \circ F, \tag{2}$$

where $g : x \to c$ is called observable and K is the Koopman operator which is an infinite linear matrix. Here, the Koopman operator advances the measurement function $g(t)$ of the state x forward in time in a linear framework as follows:

$$Kg(x_k) = g(x_{k+1}). \tag{3}$$

Identifying a subspace characterized by a finite set of measurement functions $g(t)$, where the Koopman operator remains invariant, becomes crucial at this stage.

B. Reservoir computers (RC)

Reservoir computing does indeed originate from Recurrent Neural Networks (RNNs) due to its recurrent nature. However, reservoir computing is distinguished from traditional RNNs by its unique architecture, such as the Echo State Network (ESN) or Liquid State Machine, which includes a fixed, randomly initialized reservoir of neurons and a trainable readout layer.[36] This architecture simplifies the training process compared to standard RNNs, making reservoir computing particularly suitable for tasks such as time-series prediction. In the literature, reservoir computers are implemented across diverse frameworks, encompassing photonic setups with dedicated physical hardware or agent-based reservoir computing.[37–40] The RC algorithm typically comprises three fundamental layers:

- Input Layer ($\boldsymbol{u(t)}$): Denoted by $u(t) \in \mathbb{R}^M$, representing an M-dimensional time-varying input signal.
- Reservoir Layer ($\boldsymbol{s(t)}$): Denoted by $s(t) \in \mathbb{R}^N$, consisting of a large number of neurons N. The reservoir layer plays a crucial role in capturing temporal dependencies. The reservoir connections are determined by a sparsely connected matrix W_s (of size $N \times N$), which is initially randomly chosen from a uniform distribution and remains fixed during training and testing.
- Output Layer ($\boldsymbol{y(t)}$): Denoted by $y(t) \in \mathbb{R}^K$, where K is the dimensionality of the time-varying output signal. The output layer maps the reservoir states into the K-dimensional output signal.

The input layer is associated with the input weighted matrix $W_{in}^{N\times M}$, and the reservoir layer has the reservoir weighted matrix $W_s^{N\times N}$. Initially, both W_{in} and W_s are randomly chosen from a uniform distribution. Importantly, W_s is sparsely connected, defining the reservoir connections. These matrices are fixed during both the training and testing phases. Nevertheless, the output weighted matrix W_{out} is exclusively trained to minimize the error between the actual and predicted values at the output. The reservoir states are characterized by the following equation:

$$s(t+1) = (1-a)s(t) + f\big(W_s^{N\times N}s(t) + W_{in}^{N\times M}u(t+1)\big), \tag{4}$$

where a is the leakage rate and f represents a nonlinear activation function, commonly selected as hyperbolic tangent or sigmoid. An essential parameter in reservoir computing is the leakage rate, designed to prevent over-fitting.[41] This parameter typically takes values between 0 and 1. The output of the reservoir computing is determined as follows:

$$y(t+1) = W_{out}^{K\times N}s(t+1), \tag{5}$$

where $s(t+1)$ is the reservoir states calculated in Eq. (4). Notice that the output function is obtained by the linear combination of the states multiplied by the output weights.

C. Hankel alternative view of Koopman (HAVOK)

HAVOK is an alternative approximation method to the Koopman operator. Moulder *et al.* breaks down the HAVOK algorithm into four different steps, which are preliminary attractor reconstruction, attractor decomposition into eigen time series, separation of linear ad nonlinear components, and linear representation of nonlinear dynamics.[42]

Let $g(t)$ represent the time series measurement of a nonlinear dynamical system. While the theoretical presentation focuses on a single scalar measurement ($g(t)$), it is important to note that the HAVOK framework is applicable and extends to address multiple input, multiple output (MIMO) problems as well.[34] The Hankel matrix is constructed from the time series of a measurement $g(t)$ as follows:

$$H = \begin{bmatrix} g(t_1) & g(t_2) & g(t_3) & \cdots & g(t_p) \\ g(t_2) & g(t_3) & g(t_4) & \cdots & g(t_{p+1}) \\ \vdots & \vdots & \ddots & \vdots & \\ g(t_q) & g(t_{q+1}) & g(t_{q+2}) & \cdots & g(t_n) \end{bmatrix}, \tag{6}$$

where p and q define the dimensions of Hankel matrix H and n is the size of $g(t)$. Equation (6) is equivalent to the form of the Koopman

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

operator representation,

$$H = \begin{bmatrix} g(t_1) & Kg(t_1) & K^2g(t_1) & \cdots & K^{p-1}g(t_1) \\ Kg(t_1) & K^2g(t_1) & K^3g(t_1) & \cdots & K^pg(t_1) \\ \vdots & \vdots & \ddots & \vdots & \\ K^{q-1}g(t_1) & K^qg(t_1) & K^{q+1}g(t_1) & \cdots & K^{n-1}g(t_1) \end{bmatrix}, \quad (7)$$

where K presents the Koopman operator. The application of Singular Value Decomposition (SVD) to matrix H allows for the creation of a finite-dimensional linear space. This space provides an effective approximation to the infinite-dimensional linear space mapped by the operator K. The decomposed matrix is obtained as

$$H = U\Sigma V^T. \quad (8)$$

The matrix V^T contains the principal component trajectories (PCTs) of H. Utilizing the Singular Value Decomposition of the matrix H enables the reconstruction of an accurate model for the complete dynamics. The most significant information of H matrix is contained in the columns of U and V matrices from the SVD. A further approximation to the H matrix can be obtained by considering the first r columns of V. Additionally, the first $r-1$ variables of the V matrix represent a linear model, while V_r, called as forcing term, introduces non-linearity to the model which is expressed as follows:

$$\frac{d}{dt}v(t) = Av(t) + Bv_r(t), \quad (9)$$

where $v = [\, v_1 \; v_2 \; \cdots \; v_{r-1} \,]^T$ is the vector of the eigen time-delay coordinates. A and B matrices represent the linear and forcing component of the time series measurement of a dynamical system, respectively.

V. METHODOLOGY: RC-HAVOK MODEL

Consider the dynamical system given in Eq. (1), and assume that $g(t)$ is the single measurement of the state x for time t. The RC-HAVOK method initially presents the measurement function of $g(t)$ as the input of the reservoir. Then, the internal states of the reservoir are obtained as

$$s(t+1) = (1-a)s(t) + f(W_s s(t) + W_{in} g(t+1)), \quad (10)$$

where W_{in} and W_s are decided randomly uniformly distributed weights, a is the leakage rate, and f is the nonlinear hyperbolic tangent function.

The first internal state of the reservoir is then utilized to construct the Hankel matrix for implementing the HAVOK part of the model. The eigentime delay coordinates of the internal states are obtained by performing the Singular Value Decomposition of the Hankel matrix, as indicated in Eq. (11),

$$H = \begin{bmatrix} s(t_1) & s(t_2) & s(t_3) & \cdots & s(t_p) \\ s(t_2) & s(t_3) & s(t_4) & \cdots & s(t_{p+1}) \\ \vdots & \vdots & \ddots & \vdots & \\ s(t_q) & s(t_{q+1}) & s(t_{q+2}) & \cdots & s(t_n) \end{bmatrix} = U\Sigma V^T. \quad (11)$$

Linear representation of the measurement function $g(t)$ is then obtained by the low-rank approximation using the first $r-1$ columns of the V^T. The expression is

$$\frac{d}{dt}v(t) = Av(t) + Bv_r(t), \quad (12)$$

where $v = [\, v_1 \; v_2 \; \cdots \; v_{r-1} \,]^T$ presents the eigen time-delay coordinates. Equation (12) yields the linear model of a specified dynamical system, featuring a linear A matrix and the forcing term with B matrix. Architecture of the proposed RC-HAVOK algorithm is depicted in Fig. 1. The algorithm flows as follows:

- Measure the time series data (x) of a dynamical system.
- Use the measurement data (x) as input to the RC.
- Take the data obtained from the internal states of the reservoir for delay coordinates.
- Decompose the H matrix using SVD.
- Obtain the linear (matrix A) and forcing term (matrix B) by low-rank approximation.

The process of obtaining the Hankel matrix and performing SVD is concisely outlined in Algorithm A. After deriving the linear approximation of the dynamical system, the error margin between the original Hankel matrix elements and the elements of the approximated $dv(t)$ matrix obtained in Eq. (12) is measured by using the Root Mean Square Error (RMSE).

ALGORITHM A. HAVOK with reservoir computing (RC-HAVOK).

Input: Use the internal states of the reservoir computing ($s(t)$)
Initialize **H** as a matrix of zeros with dimensions (p, size(s(t), 1) - p);
  for k from 1 to p:
  Set H[k, :] to the segment of s(t) from k to index (end-p-1+k);
  end
[U, S, V] = SVD (H, 'econ');
Compute **dv(t)** as the derivatives of **V**
Obtain matrices **A** and **B** from **dv(t)**.
error = rmse (elements of **H**, elements of **dv(t)**)

VI. NUMERICAL EXPERIMENTS AND RESULTS

A. Data generation and parameters

Parameters of the Reservoir Computing are outlined in Table I. Input weights (W_{in}) of the RC are randomly assigned values within the range (−1,1) via a uniform distribution. Similarly, the internal weights (W_r) of the reservoir are obtained using a uniform random distribution, with sparsity achieved by a connection rate of 0.200. An essential parameter for preventing over-fitting in the RC is the leakage rate, typically ranging between 0 and 1, set at 0.400. To maintain the Echo State Network criterion, the maximum absolute eigenvalue (spectral radius) of the ESN is constrained to 0.9.

In the implementation of the RC-HAVOK algorithm, time series measurements from various dynamical systems are utilized. For each dynamical system, 200 000 time series data points are fed into the RC model with a time interval of 0.001 s, spanning from 0.001 to 200 s. The train fraction is set to 1, indicating that all input data is utilized for training. Notably, the RC algorithm is not employed for prediction in this context. Instead, the internal states

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

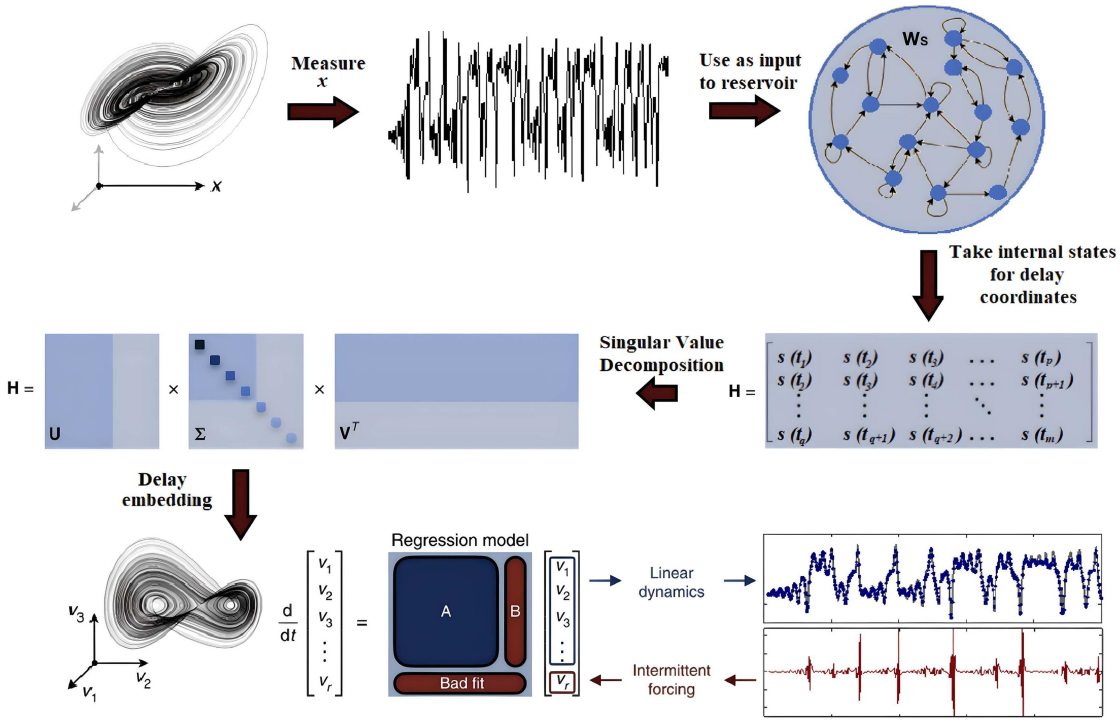


FIG. 1. Architecture of the RC-HAVOK algorithm. The algorithm is inspired by the paper written by Brunton *et al.*[34]

of the reservoir are extracted without training the model. The first internal state is then used to feed the HAVOK algorithm, facilitating the derivation of a linearized model for the dynamical systems.

B. Dynamical systems

1. Lorenz system

The Lorenz chaotic system, introduced by Edward N. Lorenz in 1963, stands as one of the pioneering examples of chaotic systems.[43]

TABLE I. Parameters of the Reservoir Computing Algorithm.

Parameter type	Value
Input weights (W_{in})	$2\times$rand(0, 1)-1
Internal weights (W_r)	rand(0. 1)
Input data	200 000
Train fraction	1
Spectral radius	0.9
Connectivity	0.200
Leakage rate	0.400
Input scaling	[0.1;0.1]
Input shift	[0;0]
Reservoir activation function	tanh
Output activation function	Identity

This third-order autonomous system exhibits intricate dynamic behavior, characterized by two quadratic terms. The differential equations governing this system are defined as follows:

$$\begin{aligned} \dot{x} &= \sigma(y - x), \\ \dot{y} &= x(\rho - z) - y, \\ \dot{z} &= xy - \beta z, \end{aligned} \tag{13}$$

where σ, β, and ρ are the system parameters and x, y, and z are the state variables of this system. When the system parameters are set as $(\sigma,\beta,\rho) = (10, 8/3, 28)$ and the initial values of the state variables are set as $(x_0, y_0, z_0) = (8, -8, 27)$, the system displays chaotic behavior.

2. Lorenz-like models: Liu system

A novel system, introduced by Liu Ling in 2004, is autonomous and operates in three dimensions.[44] This system incorporates multipliers and quadratic terms to introduce nonlinearity for folding trajectories. The output of this system exhibits highly complex chaotic dynamics, characterized by a butterfly shaped chaotic attractor structure. The differential equations governing the system are depicted as

$$\begin{aligned} \dot{x} &= \alpha(y - x) \\ \dot{y} &= bx - kxz \\ \dot{z} &= -cz + hx^2, \end{aligned} \tag{14}$$

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

where a, b, c, h, and k are the system parameters and x, y, and z are the state variable of this chaotic system. To demonstrate chaotic behavior in the system, specific system parameters are chosen: $(a, b, c, h, k) = (10, 40, 2.5, 4, 1)$ and initial state variables are set as $(x_0, y_0, z_0) = (2.2, 2.4, 38)$.

3. Lorenz-like models: Liu-Ling

In 2006, Liu and collaborators introduced the Liu-Ling model, a three-dimensional autonomous system.[45] Theoretical and numerical analyses of this system reveal intricate chaotic behaviors. In contrast to Eq. (14), the attractor structure in the Liu-Ling system takes the form of a reverse butterfly attractor. The equations governing this new structure are formulated as follows:

$$\begin{aligned} \dot{x} &= \alpha(y - x), \\ \dot{y} &= bx + kxz, \\ \dot{z} &= -cz - hxy, \end{aligned} \tag{15}$$

where a, b, c, h, and k parameters refer to the system parameters and x, y, and z parameters refer to the state variables, respectively. In this instance, to observe the presence of chaotic behavior in this system, the system parameters are represented as $(a, b, c, h, k) = (10, 40, 2.5, 4, 1)$ and initial conditions of the state variables as $(x_0, y_0, z_0) = (2.2, 2.2, 0)$.

4. Cubic Zhang system

Zhang and Han suggested a chaotic system in 2016 by introducing cubic non-linearity.[46] The novel model represented by Eq. (16) holds rich dynamics and demonstrates single-scroll, double-scroll, and four-scroll attractors,

$$\begin{aligned} \dot{x} &= byz - ax, \\ \dot{y} &= dxz - cy^3, \\ \dot{z} &= -fxy + ez, \end{aligned} \tag{16}$$

where a, b, c, d, e, and f are the system parameters and x, y, and z are the state variables of the dynamical system. When the system parameters equalize to $(a, b, c, d, e, f) = (2, 10, 6, 3, 3, 1)$ and the initial conditions of the state variables are considered as $(x_0, y_0, z_0) = (0.1, 0.1, 0.1)$, then Eq. (16) demonstrates chaotic behavior.

5. Chua circuit with PWL function

The Chua's circuit, or Chua's system, is a well-known example of a chaotic system that is a third-order ordinary differential equation system exhibiting chaotic behavior.[47] Chua's circuit was originally implemented as an electronic circuit, but its mathematical description has been used to analyze chaotic phenomena. The most well-known Chua circuit is given in Eq. (17) described by a piece-wise linear function,[48]

$$\begin{aligned} \dot{x} &= \alpha(y - h(x)), \\ \dot{y} &= x - y + z, \\ \dot{z} &= -\beta y, \end{aligned} \tag{17}$$

where

$$h(x) = h_1 x + \frac{1}{2}(h_0 - h_1)(|x + 1| - |x - 1|).$$

Note that the PWL function $h(x)$ defines

$$h(x) = \begin{cases} h_1 x + (h_0 - h_1), & x \geq 1, \\ h_0 x, & |x| \leq 1, \\ h_1 x - (h_0 - h_1), & x \leq -1. \end{cases} \tag{18}$$

Piece-wise linear expression of the Chua circuit demonstrates chaotic behavior by describing the initial conditions $(x_0, y_0, z_0) = (-0.1, 0.2, 0.1)$ and system parameters $(\alpha, \beta, h_0, h_1)$ as $(9, 100/7, -1/7, 2/7)$, respectively.

6. Chua circuit with hyperbolic tangent function

The hyperbolic tangent function ($tanh$) is often employed in the piece-wise linear resistor characteristic of Chua's circuit to introduce non-linearity.[49] The general form of the Chua circuit equations with the hyperbolic tangent function is represented as follows:

$$\begin{aligned} \dot{x} &= \alpha(y - h(x)), \\ \dot{y} &= x - y + z, \\ \dot{z} &= -\beta y, \end{aligned} \tag{19}$$

where

$$h(x) = ax - btanh(x^3)$$

and (α, β, a, b) are the system parameters with the values of $(8, 12, 0.3, 0.6)$, respectively. The initial conditions of the system are defined as $(x_0, y_0, z_0) = (0.5, 0, 0)$. The hyperbolic tangent function contributes to the generation of chaotic dynamics within the system based on the specified system parameters and initial conditions.

7. Chua circuit with cubic non-linearity

The cubic Chua system refers to an extension of Chua's circuit that includes cubic terms in the equations yielded in Eq. (20).[50] This modification introduces additional nonlinearities to the system and thus enhances the complexity,

$$\begin{aligned} \dot{x} &= \alpha(y - h(x)), \\ \dot{y} &= x - yz, \\ \dot{z} &= -\beta y, \end{aligned} \tag{20}$$

where

$$h(x) = \frac{(2x^3 - x)}{7}$$

and $\alpha = 10$ and $\beta = 100/7$. Equation (20) exhibits chaotic dynamics using the specified parameters and initial conditions of $(x_0, y_0, z_0) = (-0.3, 0.5, 0.4)$.

C. RC-HAVOK modeling of dynamical systems with different non-linearities

The RC-HAVOK algorithm is applied to various datasets generated from dynamical systems with different types of nonlinearities.

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

Specifically, cubic non-linearity, hyperbolic tangent function, and piece-wise linear function are investigated to approximate the Koopman operator. The modeling results for the cubic Zhang system and the cubic Chua system are presented in Secs. VI C 1 and VI C 2, respectively. Additionally, results for the hyperbolic tangent function and PWL function forms of the Chua system are introduced in Secs. VI C 3 and VI C 4, respectively. The specific HAVOK parameters employed for implementing the example dynamical systems are detailed in the Appendix.

1. RC-HAVOK representation of the cubic zhang system

The RC-HAVOK algorithm is employed to investigate the implementation feasibility of the cubic Zhang chaotic system described in Eq. (16). To assess the impact of the reservoir computer, the cubic Zhang system is initially modeled solely using the HAVOK algorithm and subsequently approximated through the RC-HAVOK approach. Implementation results, including rank size, error rate, and modeling time for various reservoir structures, are provided in Table II.

TABLE II. RC-HAVOK modeling of the cubic Zhang time series for different numbers of neurons in the reservoir structure.

	Rank size	Error rate (*rmse*)	Modeling time (s)
Without reservoir	r = 17	e = 0.0013	t = 8.5
N = 0	r = 6	e = 0.0025	t = 6.31
With reservoir	r = 8	e = 0.0013	t = 7.8
N = 300	r = 6	e = 0.0016	t = 7.5
With reservoir	r = 6	$e = 1.7875 \times 10^{-4}$	t = 8.74
N = 500	r = 4	$e = 2.0122 \times 10^{-4}$	t = 8.9738

The impact of the RC-HAVOK algorithm is evident in the results presented in Table II. The linear HAVOK model of the cubic Zhang system (without reservoir) exhibits a rank size of 17 with an rmse rate of 0.0013. This implies that the linear HAVOK approximation of Eq. (16) can be represented by 17 × 17 matrix as defined in the architecture displayed in Fig. 1. However, thanks to the RC-HAVOK algorithm, it is possible to reduce the rank size from 17

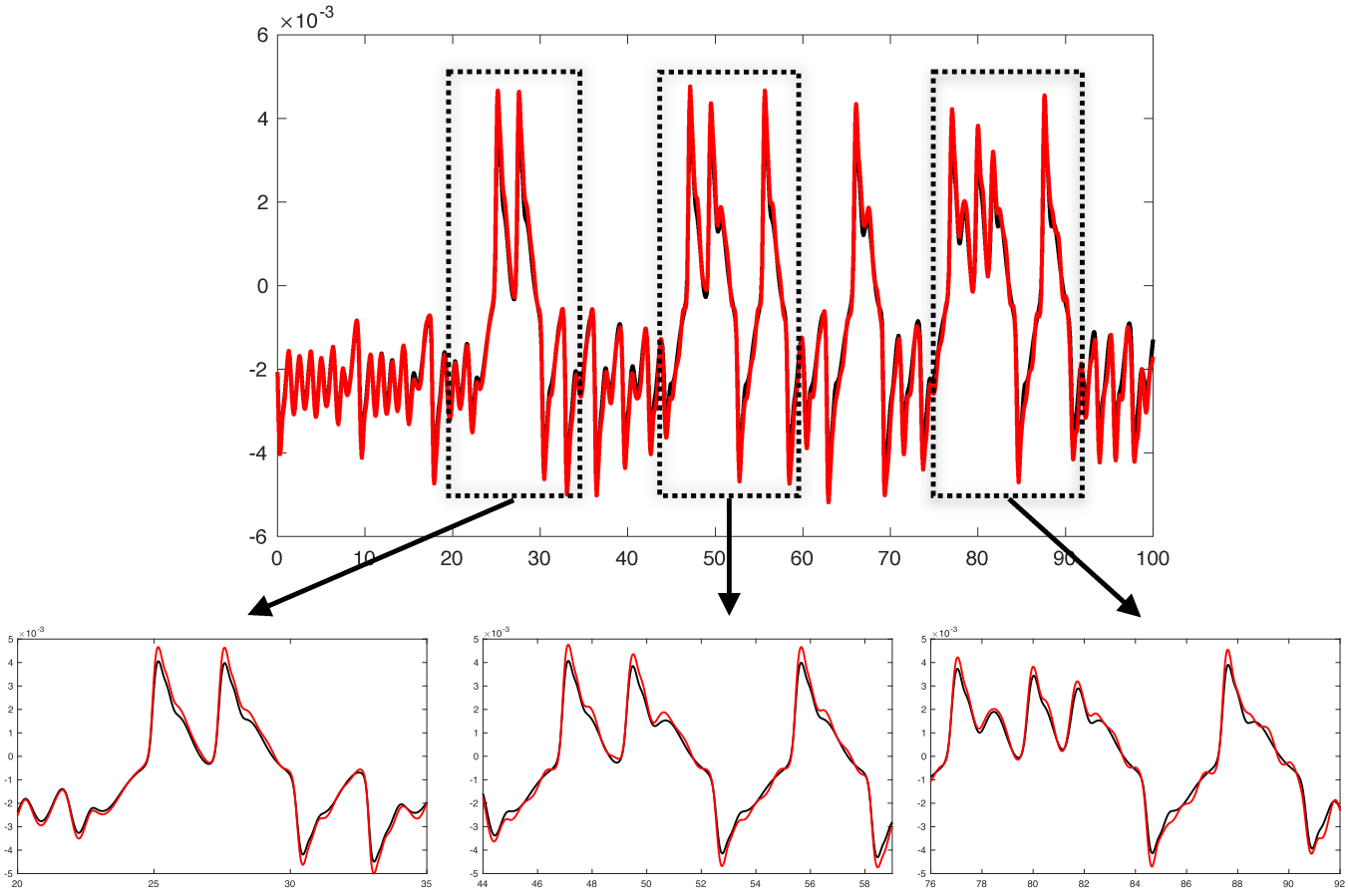


FIG. 2. Time series data from the RC-HAVOK implementation (red) compared to the time-delay coordinates (black) of the cubic Zhang dynamical system.

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

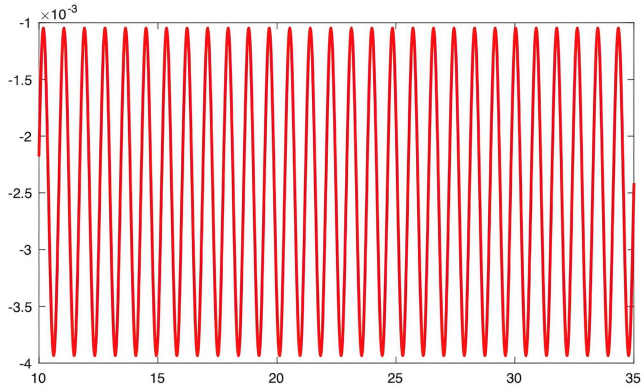


FIG. 3. RC-HAVOK model of the cubic Zhang system without forcing term.

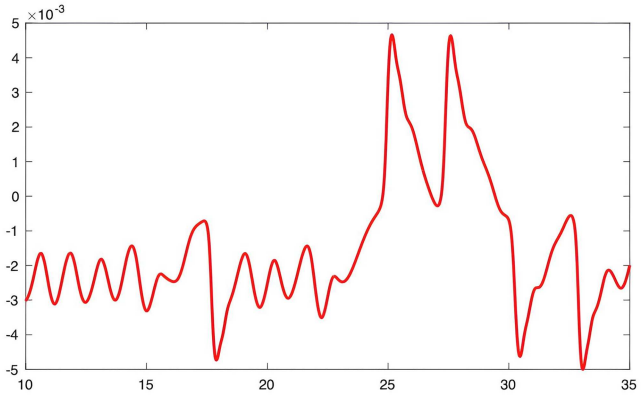


FIG. 4. RC-HAVOK model of the cubic Zhang system with forcing term.

to 4 and represents the linear approximation of Eq. (16) with just a 4×4 matrix. Furthermore, the error rate is significantly improved with introduction of the RC-HAVOK approach. Consequently, the minimum rank size for the linear cubic Zhang system is determined as 4, and subsequent analyses are carried out based on this value.

The time series of the cubic Zhang system, derived from the time-delay coordinates of the SVD and the linear RC-HAVOK algorithm, is depicted in Fig. 2. The graph is zoomed in at three different time periods for better understanding, each revealing a close alignment between the linear RC-HAVOK time series and the time-delay coordinates.

Figures 3 and 4 demonstrate the time series of the linear RC-HAVOK for cubic Zhang system. In Eq. (9), the $v(t)$ vector with $r-1$ variables represents the linear part, while v_r, referred to as the forcing term, is the nonlinear part of the model. Figure 3 is the evidence of the linear representation of the RC-HAVOK algorithm by setting the forcing term to zero. By implementing this adjustment, the influence of the forcing term is eliminated, allowing for the observation of the pure linear model. On the other hand, the forcing term is included in Fig. 4, which exhibits the complete linear RC-HAVOK model for the cubic Zhang system.

Figure 5(a) depicts the cubic Zhang attractor, generated using time-delay coordinates obtained from the Hankel matrix. It also illustrates the outcomes of both the linear RC-HAVOK algorithm and the linear HAVOK algorithm, respectively. The output of the RC-HAVOK algorithm closely aligns with the original time-delay embedded attractor, while the output of the HAVOK algorithm diverges significantly from the original attractor. Consequently, the enhancement achieved by the RC-HAVOK algorithm is clearly revealed through the utilization of the approximated attractor of the linear cubic Zhang system.

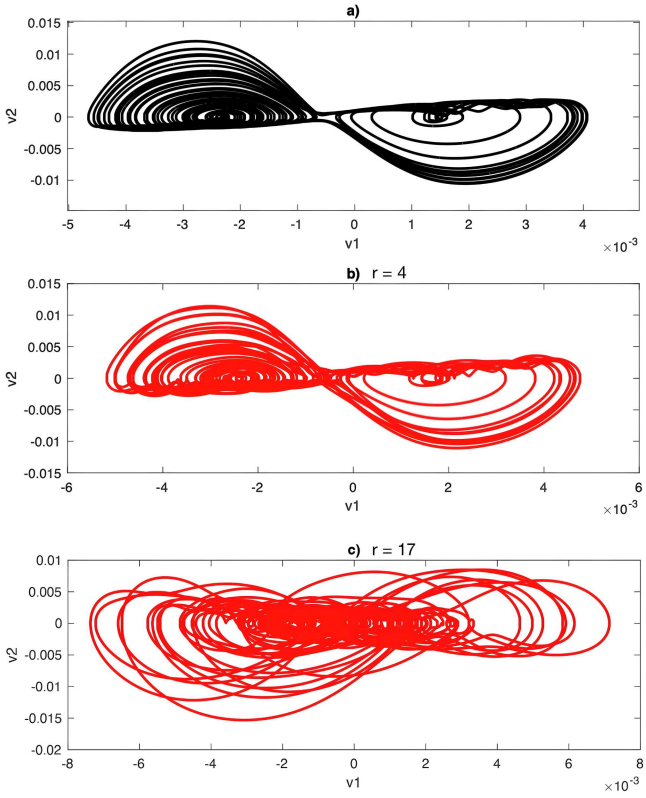


FIG. 5. (a) Time-delay embedded attractor of the cubic Zhang system using the time-delay coordinates coming from the singular value decomposition of the Hankel matrix as described in Eq. (11). (b) Reconstructed attractor of the cubic Zhang system using the linear RC-HAVOK model with forcing term $v_r(r = 4)$. (c) Reconstructed attractor of the cubic Zhang system using the linear HAVOK model with forcing term $v_r(r = 17)$.

2. RC-HAVOK representation of the Chua circuit with cubic non-linearity

Cubic Chua circuit is another example of the well-known dynamical systems with cubic non-linearity as described in Eq. (20). The linear RC-HAVOK model of this system is obtained, and

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

TABLE III. RC-HAVOK modeling of the cubic Chua time series for different numbers of neurons in the reservoir structure.

	Rank size	Error rate (*rmse*)	Modeling time (s)
N = 0 (W/o reservoir)	r = 6	e = 0.0017	t = 5.94
N = 200	r = 4	e = 0.0015	t = 6.8
N = 300	r = 4	e = 0.0015	t = 7.2
N = 500	r = 4	e = 0.0014	t = 8.5

the results are presented in Table III. The RC-HAVOK algorithm exhibits the advantage of a lower rank size and smaller error rate compared to the classical HAVOK algorithm (when $n = 0$). However, this advantage of the RC-HAVOK results in a slight increase in the modeling time. Due to having both a lower rank size and error rate, the RC-HAVOK representation of the cubic Chua circuit is realized based on the reservoir structure with $n = 500$ neurons.

Figure 6 demonstrates the three-dimensional phase-space representation of the time-delay embedded attractor using the time-delay coordinates of the cubic Chua circuit. On the other hand, Fig. 7 presents the reconstructed attractor obtained by the linear RC-HAVOK model for the cubic Chua circuit. This attractor closely resembles the time-delay embedding attractor demonstrated in Fig. 6. These results serve as evidence of the successful implementation of the RC-HAVOK approach on the cubic Chua circuit.

3. RC-HAVOK representation of the Chua circuit with hyperbolic tangent function

To validate the feasibility of the RC-HAVOK algorithm across various nonlinearities, a modified version of the Chua circuit incorporating a hyperbolic tangent function is employed. Initially, the dynamics outlined in Eq. (19) are measured, and the resulting

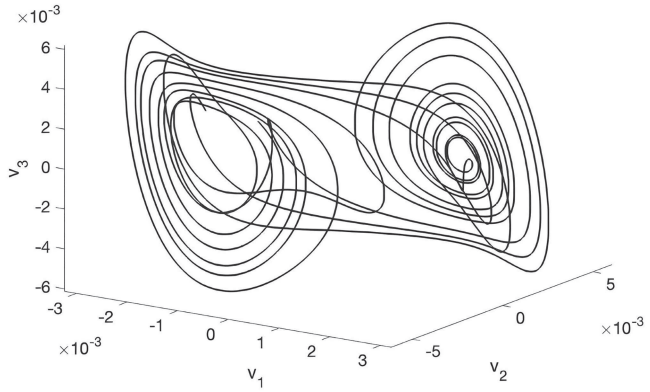


FIG. 6. Time-delay embedded attractor of the cubic Chua circuit using the time-delay coordinates coming from the singular value decomposition of the Hankel matrix for the cubic Chua circuit.

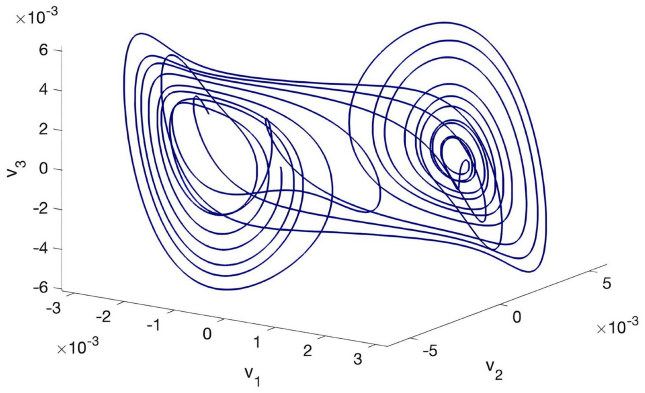


FIG. 7. Reconstructed attractor of the cubic Chua circuit using the linear RC-HAVOK model with forcing term $v_r (r = 4)$ for the cubic Chua circuit.

time series data is applied to the RC-HAVOK algorithm. The implementation results for varying rank sizes are presented in Table IV. For clearer insight, the hyperbolic Chua circuit is modeled using the HAVOK (the case $n = 0$) and RC-HAVOK (the case $n = 500$) algorithms for the same rank sizes. Table IV obviously illustrates the superiority of the RC-HAVOK approach over the traditional HAVOK algorithm based on the error rates of the models. The optimum linear RC-HAVOK model of the hyperbolic Chua circuit is obtained when the rank size equals 4. Therefore, the RC-HAVOK representation of the Eq. (19) is performed on this value.

Figure 8 displays the time series graphs for the three different cases. Figure 8(a) represents the time-delay coordinates of the hyperbolic Chua circuit, serving as a reference to assess the accuracy of the linear models. Figures 8(b) and 8(c) depict the RC-HAVOK and HAVOK representations of the hyperbolic Chua circuit, respectively. Both linear models are observed with a defined rank size of $r = 4$. The time series graph in Fig. 8 clearly illustrates that the RC-HAVOK approach provides the best approximation to the reference time series, while the traditional HAVOK algorithm leads to completely different behaviors over time.

TABLE IV. RC-HAVOK modeling of the Chua circuit with hyperbolic tangent function for different numbers of rank sizes.

	Rank size	Error rate (*rmse*)	Modeling time (s)
Without reservoir N = 0	r = 10	e = 0.0024	t = 8.2
	r = 8	e = 0.0021	t = 8.1
	r = 6	e = 0.0029	t = 8.0
	r = 4	e = 0.0052	t = 7.7
With reservoir N = 500	r = 10	$e = 3.05 \times 10^{-4}$	t = 12.53
	r = 8	$e = 2.95 \times 10^{-4}$	t = 11.85
	r = 6	$e = 3.45 \times 10^{-4}$	t = 10.13
	r = 4	$e = 2.64 \times 10^{-4}$	t = 9.6

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

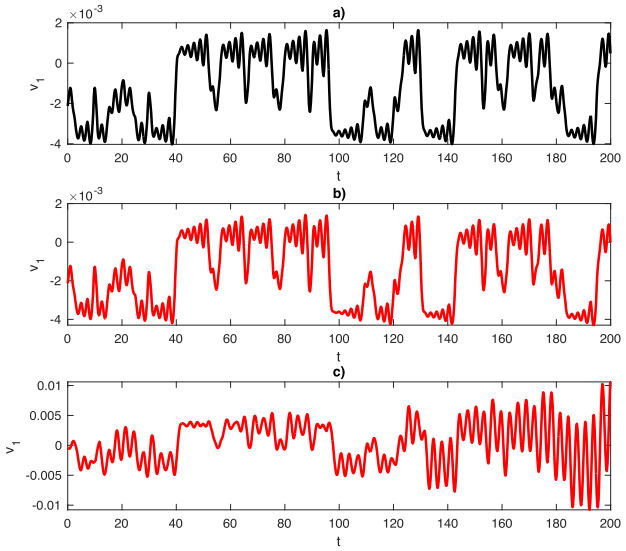


FIG. 8. (a) Time-delay coordinates of the hyperbolic Chua circuit coming from the singular value decomposition of the Hankel matrix. (b) Reconstructed time series of the hyperbolic Chua circuit using the linear RC-HAVOK model with forcing term $v_r(r = 4)$. (c) Reconstructed time series of the hyperbolic Chua circuit using the linear HAVOK model with forcing term $v_r(r = 4)$.

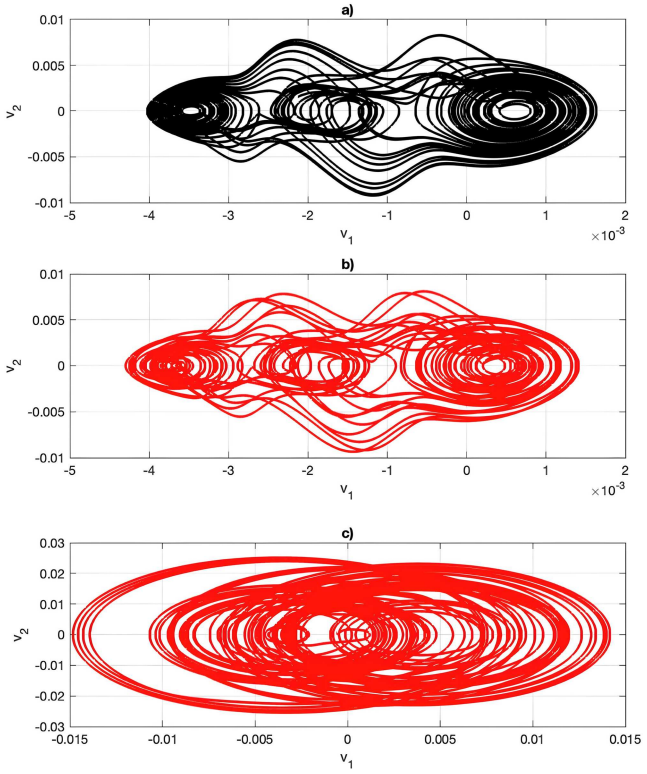


FIG. 9. (a) Time-delay embedded attractor of the hyperbolic Chua circuit using the time-delay coordinates coming from the singular value decomposition of the Hankel matrix. (b) Reconstructed attractor of the hyperbolic Chua circuit using the linear RC-HAVOK model with forcing term $v_r(r = 4)$. (c) Reconstructed attractor of the hyperbolic Chua circuit using the linear HAVOK model with forcing term $v_r(r = 4)$.

The assertion regarding the "completely different behavior" of the HAVOK model, as mentioned in the preceding paragraph, is elucidated by observing Fig. 9. The attractors presented in Fig. 9 correspond to the attractors of the time series obtained in Fig. 8. The best approximation to the time-delay embedded attractor, demostrated in Fig. 9(a), is achieved through the RC-HAVOK approach. The linear reconstructed attractor of the hyperbolic Chua circuit based on the RC-HAVOK is displayed in Fig. 9(b). In contrast, the reconstructed attractor of the linear HAVOK model is depicted in Fig. 9(c). The shape of this attractor differs from the one presented in Figs. 9(a) and 9(b), owing to disparities in the time series data.

4. RC-HAVOK representation of the Chua circuit with PWL function

Chua circuit is initially constructed by using the piece-wise linear function as defined in Eq. (17). The RC-HAVOK implementation of this dynamical system is investigated and the results are introduced in Table V.

Table V displays the linear approximation of the Chua circuit with a PWL function using two approaches: the HAVOK algorithm ($n = 0$) and the RC-HAVOK algorithm ($n = 500$). The error rates of the linear model for various rank sizes are significantly high for the best approximation in both cases. Despite these relatively high error rates, the RC-HAVOK approach still outperforms the HAVOK algorithm. The linear RC-HAVOK model for the Chua circuit with PWL function is obtained with a rank size of 6, reflecting a lower error rate.

TABLE V. RC-HAVOK modeling of the Chua circuit with PWL function for different numbers of rank sizes.

	Rank size	Error rate (*rmse*)	Modeling time (s)
Without reservoir N = 0	r = 10	e = 0.0040	t = 6.16
	r = 8	e = 0.0040	t = 6.10
	r = 6	e = 0.0035	t = 6.04
	r = 4	e = 0.0067	t = 5.98
With reservoir N = 500	r = 10	e = 0.0036	t = 9.77
	r = 8	e = 0.0035	t = 9.72
	r = 6	e = 0.0028	t = 8.83
	r = 4	e = 0.0050	t = 8.95

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

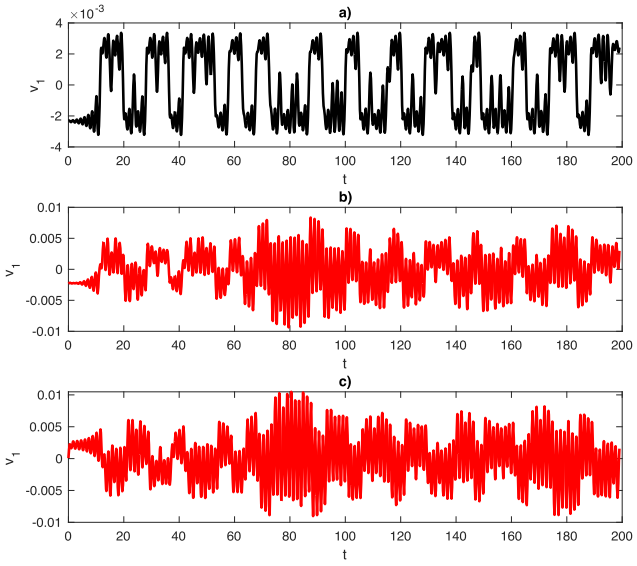


FIG. 10. (a) Time-delay coordinates of the PWL Chua circuit coming from the singular value decomposition of the Hankel matrix. (b) Reconstructed time series of the PWL Chua circuit using the linear RC-HAVOK model with forcing term $v_r(r = 6)$. (c) Reconstructed time series of the PWL Chua circuit using the linear HAVOK model with forcing term $v_r(r = 6)$.

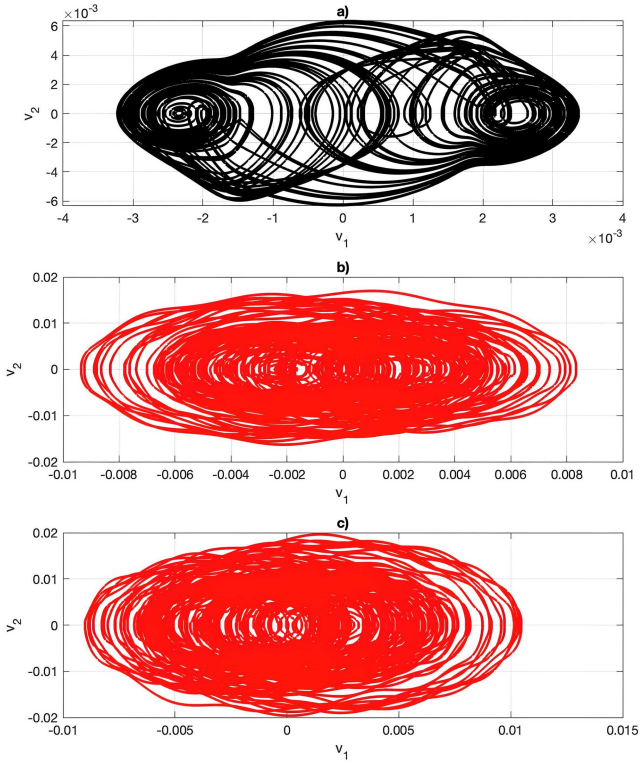


FIG. 11. (a) Time-delay embedded attractor of the PWL Chua circuit using the time-delay coordinates coming from the singular value decomposition of the Hankel matrix. (b) Reconstructed attractor of the PWL Chua circuit using the linear RC-HAVOK model with forcing term $v_r(r = 6)$. (c) Reconstructed attractor of the PWL Chua circuit using the linear HAVOK model with forcing term $v_r(r = 6)$.

Figure 10 illustrates the time series data derived from the time-delay coordinates, the RC-HAVOK model and the HAVOK model of the Chua circuit with PWL function, respectively. The adverse impact of the higher error rate is obviously observed from the time series data of the linear models. The slight divergence in the linear approximation results in the observation of distinct dynamics in the RC-HAVOK and HAVOK algorithms.

The distinct dynamics of the PWL Chua circuit, as observed in the times series of the RC-HAVOK and HAVOK models in Fig. 10, are further clarified through the attractors of these models. Figure 11(a) illustrates the time-delay embedded attractor of the PWL Chua circuit. For a more accurate approximation, reconstructed attractors of the RC-HAVOK and HAVOK models should exhibit a similar structure to the time-delay embedded attractor. Nevertheless, the structure of the attractors depicted in Figs. 11(b) and 11(c) significantly differ from the attractor presented in Fig. 11(a). This scenario highlights the sensitivity of dynamical systems, where minor alterations can result in substantial deviations.

D. RC-HAVOK modeling of Lorenz-like dynamical systems

To assess the impact of the RC-HAVOK algorithm, it is important to apply the approach to various systems. This section investigates the feasibility of the RC-HAVOK algorithm on Lorenz-like dynamical systems. Modeling results for these systems, including the quadratic Lorenz, Liu and Liu-Ling systems are presented in Secs. VI D 1–VI D 3, respectively. The detailed HAVOK parameters used for implementing the Lorenz-like dynamical systems can be found in the Appendix.

1. Lorenz quadratic system

The first system examined is the quadratic Lorenz structure. In this case, the quadratic Lorenz system is initially constructed using the HAVOK algorithm and subsequently modeled with the inclusion of the RC-HAVOK approach. Implementation results, including rank size, error rate, and modeling time for different reservoir structures, are presented in Table X.

To comprehend the impact of the RC-HAVOK method on the Lorenz system, it is essential to review Table X. Notably, when the number of neurons is set to zero, both the matrix rank and error rate exhibit high values. The advantage of such a configuration lies in the system's ability to generate outputs in a shorter time frame. Conversely, when the number of neurons in the reservoir structure is increased to 300, along with the incorporation of reservoir computing, there is a noticeable reduction in error rates and matrix rank sizes. However, this reduction is accompanied by an extended

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

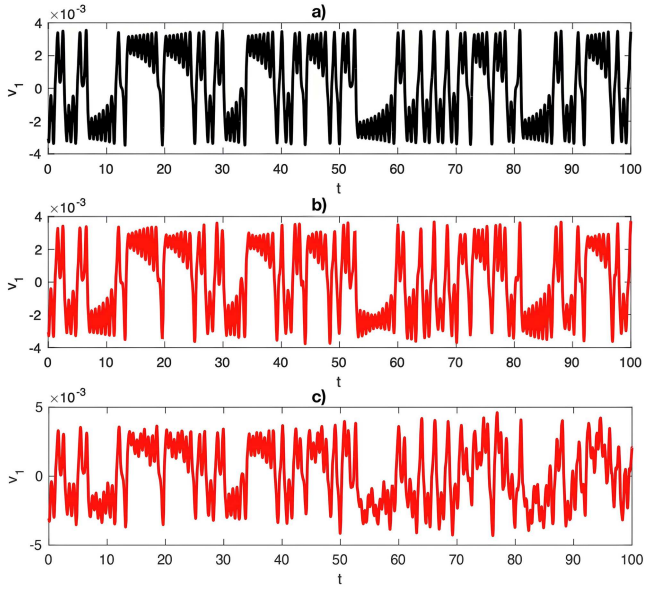


FIG. 12. Time series of the quadratic Lorenz system: (a) Coming from the Singular Value Decomposition of the Hankel matrix. (b) The linear RC-HAVOK model with forcing term $v_r(r = 4)$. (c) The linear HAVOK model with forcing term $v_r(r = 4)$.

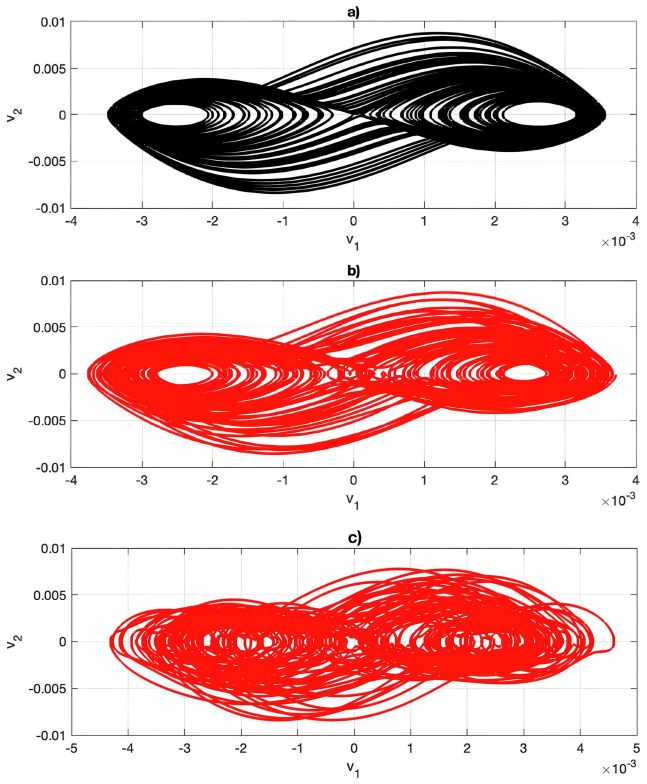


FIG. 13. (a) Time-delay embedded attractor of the quadratic Lorenz system using the time-delay coordinates coming from the singular value decomposition of the Hankel matrix. (b) Reconstructed attractor of the quadratic Lorenz system using the linear RC-HAVOK model with forcing term $v_r(r = 4)$. (c) Reconstructed attractor of the quadratic Lorenz system using the linear HAVOK model with forcing term $v_r(r = 4)$.

runtime, attributed to the increased complexity of the algorithm. For example, in computations excluding the reservoir, a matrix rank size (r) of 15 and a root mean square error ($rmse$) of 0.0026 result in a modeling time (t) of 8.1 s. In contrast, the optimal matrix rank size and error rate for the Lorenz system are determined to be 4 and 2×10^{-4}, respectively.

The implementation results of the RC-HAVOK algorithm on the quadratic Lorenz system are depicted in Figs. 12 and 13. The time series of the delay coordinates and the RC-HAVOK model for the Lorenz system is illustrated in Fig. 12. The outcomes of the linear RC-HAVOK model closely match the original time series derived from the Hankel matrix. This alignment demonstrates an accurate linear RC-HAVOK approximation of the quadratic Lorenz system.

The accurate linear approximation for the Lorenz system is further confirmed by examining the time-delay embedded attractor and the RC-HAVOK attractor, as demonstrated in Figs. 13(a) and 13(b), respectively. The small error rate results in the attainment of the expected attractor structure, and it is noted that the two structures closely match. The distinct benefit of the RC-HAVOK approach becomes apparent with the reduction of the rank size from 15 to 4 for the quadratic Lorenz system, when compared to the conventional HAVOK algorithm.[34]

The RC-HAVOK algorithm initially processes the input data through a reservoir. The reservoir structure is generated randomly with values in the range of $(-1, 1)$ using a uniform distribution. This randomness might seem to influence the subsequent steps of the algorithm. To demonstrate the reproducibility of the RC-HAVOK algorithm's modeling success, we implemented the RC-HAVOK model on the quadratic Lorenz system 50 times. For each iteration, we measured the error rates and modeling time, then calculated the minimum, average, and maximum error rates to quantify the degree of deviation. The minimum error rate over 50 iterations is 2.52×10^{-4}, and the maximum error rate is 7.20×10^{-4}. The average error rate across 50 iterations is 5.52×10^{-4}. Additionally, the minimum and maximum modeling times are measured at 11.19 and 19 s, respectively, with an average modeling time of 12.74 s across

TABLE VI. Error rates and modeling times through 50 iterations of RC-HAVOK algorithm on Lorenz system.

	Error rate ($rmse$)	Modeling time (s)
Minimum value	2.52×10^{-4}	11.19
Average value	5.52×10^{-4}	12.74
Maximum value	7.20×10^{-4}	19

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

TABLE VII. RC-HAVOK modeling of the Liu time series for different numbers of neurons in the reservoir structure.

	Rank size	Error rate (*rmse*)	Modeling time (s)
Without reservoir N = 0	r = 15	e = 0.0005	t = 8.2
With reservoir N = 300	r = 8	e = 0.0005	t = 8
With reservoir N = 300	r = 6	e = 0.0008	t = 7.6

the 50 iterations. The calculated values for error rates and modeling times are depicted in Table VI. These results demonstrate that the RC-HAVOK algorithm can reliably produce a linear approximation of the Lorenz system. Additional dynamical systems used in this study could also be utilized to further validate the reproducibility of the RC-HAVOK approximation. However, they have been excluded from this study to maintain simplicity.

2. Liu system

The evaluation of the impact of the RC-HAVOK approach on the Liu system is essential. Table VII offers insights into the system's behavior when subjected to the RC-HAVOK approach. As depicted in the table, the initial analysis considers the system without the reservoir structure. Opting for a matrix rank size of $r = 15$ results in an error value of 0.0005, and the system manifests chaotic behavior after 8.2 s. However, incorporating the reservoir structure into the system with 300 neurons reveals optimal performance with a matrix rank size of $r = 6$, yielding an error value of 0.0008, and chaotic behavior is observed after 7.6 s. Notably, in contrast to the Lorenz system, including the reservoir structure leads to a reduction in output time and an increase in the error rate.

The time series and attractors obtained through the application of the RC-HAVOK approach to the Liu chaotic system, one of the Lorenz-like systems, are depicted in Figs. 14 and 15, respectively. The overlaid time series graphs demonstrate consistent results, and this similarity is also observed in the attractor structures. The linear RC-HAVOK model of the Liu system exhibits dynamics similar to the time-delay embedded attractor of the system.

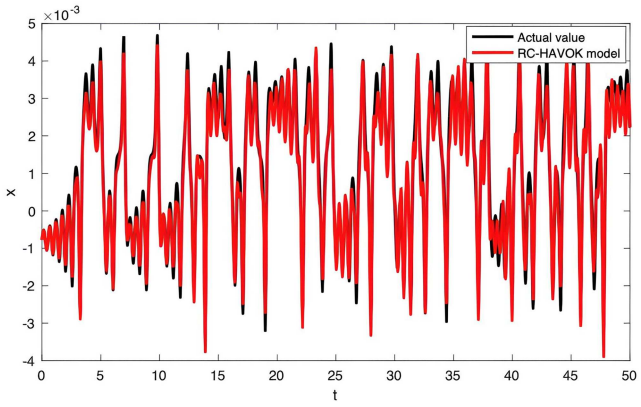


FIG. 14. Time series of the Liu system coming from the singular value decomposition of the Hankel matrix (black) and the linear RC-HAVOK model with forcing term $v_r(r = 6)$ (red).

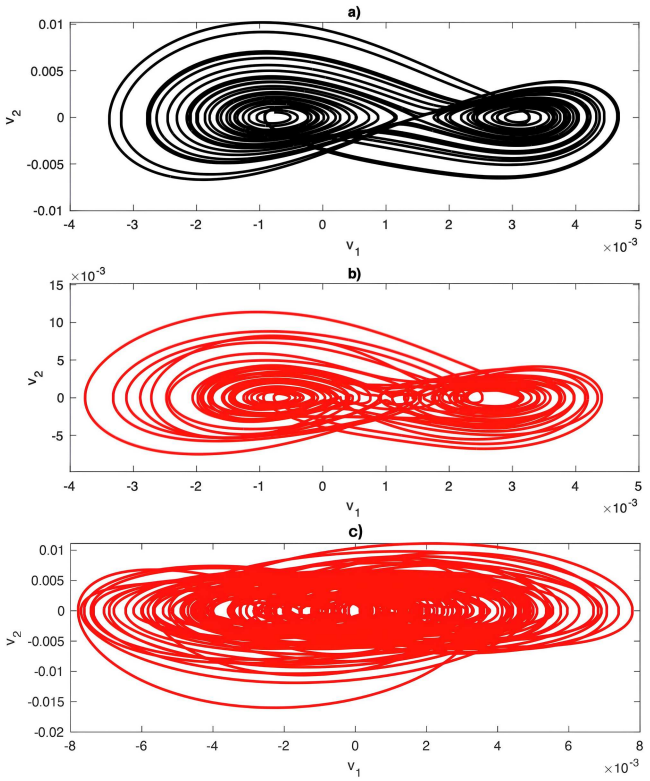


FIG. 15. (a) Time-delay embedded attractor of the Liu system using the time-delay coordinates coming from the singular value decomposition of the Hankel matrix. (b) Reconstructed attractor of the Liu system using the linear RC-HAVOK model with forcing term $v_r(r = 6)$. (c) Reconstructed attractor of the Liu system using the linear HAVOK model with forcing term $v_r(r = 6)$.

3. Liu-Ling system

The check structures of the Liu-Ling system, developed as a new approach to the Liu system, are presented as a result of

TABLE VIII. RC-HAVOK modeling of the Liu-Ling time series for different numbers of neurons in the reservoir structure.

	Rank size	Error rate (*rmse*)	Modeling time (s)
Without reservoir N = 0	r = 12	e = 0.0009	t = 7.5
With reservoir N = 300	r = 8	e = 0.0013	t = 8.3
With reservoir N = 300	r = 6	e = 0.0009	t = 7.9
With reservoir N = 300	r = 4	e = 0.0013	t = 7.8

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

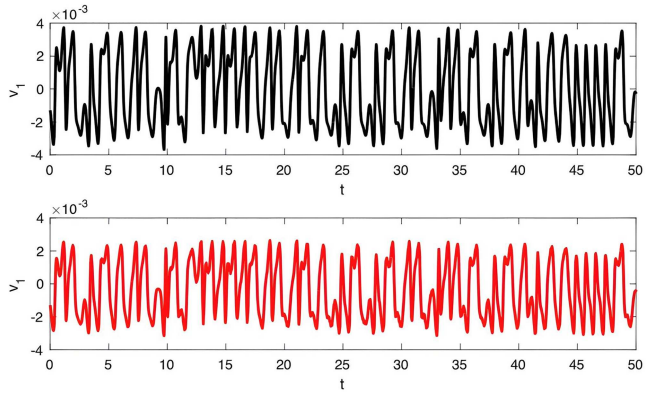


FIG. 16. Time series of the Liu-Ling system coming from the singular value decomposition of the Hankel matrix (black) and the linear RC-HAVOK model with forcing term $v_r(r = 6)$ (red).

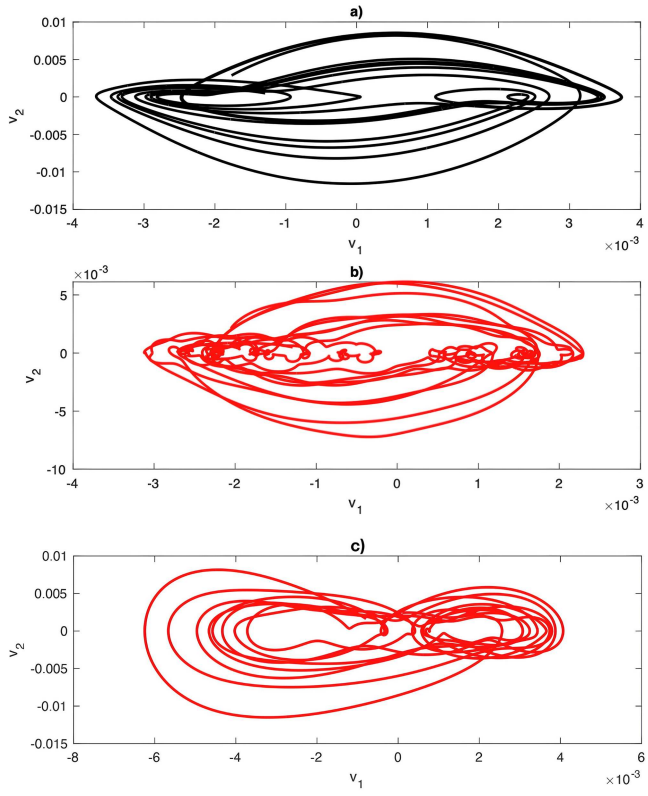


FIG. 17. (a) Time-delay embedded attractor of the Liu-Ling system using the time-delay coordinates coming from the singular value decomposition of the Hankel matrix. (b) Reconstructed attractor of the Liu-Ling system using the linear RC-HAVOK model with forcing term $v_r(r = 6)$. (c) Reconstructed attractor of the Liu-Ling system using the linear HAVOK model with forcing term $v_r(r = 6)$.

the analysis with the RC-HAVOK approach. Initially, finding the optimum value of the matrix rank size is performed using a reservoir structure with 300 neurons. The selected matrix size values, error rates, and runtimes are shown in Table VIII. As indicated in Table VIII, first, when the matrix rank size is set to $r = 12$ without including the reservoir, the error rate and modeling time are 0.0009 and 7.5 s, respectively. When the number of neurons is set to 300, and the reservoir is included, the optimum value of the matrix rank size is found to be $r = 4$, with the error rate and modeling time calculated as 0.0013 and 7.8 s, respectively. This analysis reveals that the modeling time remains almost the same even without the reservoir, while the error rate is approximately 1.5 times less than without the reservoir.

The feasibility of the RC-HAVOK approximation is observed on the Liu-Ling system through the analysis of the time series and attractor structure as displayed in Figs. 16 and 17. The chaotic behavior of the linear RC-HAVOK model closely aligns with the time-delay coordinates of the Liu-Ling system as seen in Fig. 16. This alignment is essential for creating similar attractor structures. At this point, Fig. 17 further confirms that the linear Liu-Ling model exhibits similar dynamics to the original structure.

VII. DISCUSSION

A. Comparison of the data-driven approaches: Strengths and weakness of RC-HAVOK

This paper presents a novel dimension reduction algorithm based on reservoir computers and the Koopman operator. The algorithm is specifically designed to extract essential features from a dynamical system, relying solely on the measurements of the state variables without requiring information about the system parameters or the mathematical model. One drawback of Koopman modeling is its representation of a nonlinear system within a higher-dimensional linear framework. This implies dealing with a high-dimensional linear model instead of a relatively lower-dimensional nonlinear structure. The RC-HAVOK algorithm proposed in this study addresses this limitation, aiming to model a dynamical system with reduced dimensions while minimizing the error rate. This superior advantage of the algorithm is compared to other existing studies that focus on approximating the time series of the Lorenz system given in Eq. (13), and the results are presented in Table IX.

TABLE IX. Comparison of the models focuses on approximating the time series of the Lorenz chaotic system given in Eq. (13). The lowest error rate and the lowest rank size are shown in **bold**.

Model name	Error rate	Rank size
HAVOK[31]	0.0026 rmse	r = 15
HAVOK-ML[51]	$\mathbf{1.43\times10^{-5}}$ **rmse**	r = 11
NbedDyn[52]	$6.5 \times 10^{-4} - 0.9 \times 10^{-4}$ rmse	r = 6
DLDMD[27]	1.79×10^{0} mse	N/P
RC-EDMD[24]	1.17×10^{0} nrmse	N/P
RC-HAVOK	2.4×10^{-4} rmse	**r = 4**

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

TABLE X. RC-HAVOK modeling of the Lorenz time series for different numbers of neurons in the reservoir structure.

Number of neurons in reservoir	Rank size	Error rate (*rmse*)	Modeling time (s)
N = 0 (without reservoir)	r = 15	e = 0.0026	t = 8.1
N = 100	r = 13	e = 0.0026	t = 8.03–8.16
N = 200	r = 11	e = 0.0025	t = 8.3–8.4
N = 300	r = 10	e = 0.0011	t = 8.5–8.7
N = 500	r = 4	$e = 2.4 \times 10^{-4}$	t = 9.0–9.2

TABLE XI. Rank sizes and error rates of the dynamical systems utilized in this study.

Dynamical systems	Rank size *r*		Error rate (*rmse*)	
	HAVOK	RC-HAVOK	HAVOK	RC-HAVOK
Cubic Zhang	17	4	0.0013	2.0122×10^{-4}
Cubic Chua	6	4	0.0017	0.0014
Hyperbolic Chua	8	4	0.0021	2.64×10^{-4}
Chua PWL	6	6	0.0035	0.0028
Lorenz system	15	4	0.0026	2.4×10^{-4}
Liu system	15	8	0.005	0.005
Liu-Ling system	12	6	0.009	0.009

Brunton *et al.* introduced the HAVOK algorithm to approximate the Koopman operator and applied it to various time series data, including the Lorenz chaotic system.[31] Their study demonstrated that a linear representation of the Lorenz system could be achieved with a rank size of 15, indicating the size of the approximated linear matrix derived from the HAVOK approach. The error rate between the linear approximation and the original time series was calculated as 0.0026 using the RMSE metric. Building on this foundational work, alternative approaches to the HAVOK algorithm have been proposed. Yang *et al.* introduced a hybrid method combining HAVOK with Machine Learning (ML).[51] The HAVOK-ML method simulates a closed linear model of the Lorenz system with an RMSE error of 1.43×10^{-5}, the lowest error rate reported in Table VIII. However, the linear model's size is specified as 11, which is relatively high compared to our proposed RC-HAVOK approach. Additionally, Ouala *et al.* reduced the rank size of the Lorenz system to 6 by implementing the Neural Embedding for Dynamical Systems (NbedDyn) approach.[52]

Another study combined Deep Learning (DL) with the Dynamic Mode Decomposition algorithm, modeling the Lorenz system with a higher error rate than other studies reported in the table.[27] Gulina and Mauroy conducted similar research by combining reservoir computing with the Extended Dynamic Mode Decomposition.[24] They claimed that one of their proposed methods could model the Lorenz system with a lower rank size, but they did not provide numerical values. Considering all these studies, our proposed RC-HAVOK algorithm achieves the lowest rank size for the linearized Lorenz system. The RC-HAVOK algorithm offers a significant advantage by reducing the rank size by approximately four times and improving the error rate by a factor of ten compared to the traditional HAVOK algorithm. While the HAVOK-ML method should be preferred for achieving a lower error rate, it requires working with a linear approximation of the Lorenz system that has a rank size approximately three times larger.

B. Effect of the reservoir structure

The RC-HAVOK algorithm comprises a trade-off in determining the reservoir structure, where the rank and error rate of the modeled system are notably influenced by the number of neurons in the reservoir. Table X displays the modeling results for the Lorenz time series with varying neuron (n) values. Implementation results distinctly indicate that increasing the number of neurons in the reservoir leads to a reduction in rank size and a lower error rate. However, this improvement comes at the expense of a slight increase in modeling time. Hence, it is crucial to prioritize the specifications that the model requires.

C. Effect of the different nonlinearities

The RC-HAVOK algorithm demonstrates its feasibility on various dynamical systems with different nonlinearities, as discussed in Sec. VI. The RC-HAVOK approach can provide a linear representation for all dynamical systems presented in this study, except for the Chua circuit with a PWL function. To understand this exception, we explored potential reasons. Li and Jiang suggested that sufficiently smooth activation functions are crucial for accurate approximation.[53] Additionally, Moulder *et al.* have argued that optimal HAVOK models can be achieved with smooth polynomial functions.[42] Furthermore, in addition to these studies, there are other research works that assert the necessity of having a smooth activation function for improved approximation.[26,52,54,55] This study aligns with these assessments and validates them by implementing the RC-HAVOK model on the PWL function, which is continuous but non-smooth. The error rate of the linear RC-HAVOK model for the Chua circuit with the PWL function is higher compared to the others. The shortcomings of the model are also apparent in the time series and reconstructed attractor graphs presented in Figs. 10 and 11, respectively.

While the original HAVOK algorithm effectively approximates the linear Koopman operator, our introduction of reservoir computing to the original HAVOK aims to achieve a closed linear model with reduced dimensions, referred to as the rank size. By applying this approach to a broad range of dynamical systems, we observed that the RC-HAVOK algorithm can model with a lower error rate and reduced rank size in certain cases. However, this improvement comes at the cost of increased modeling time due to the added complexity of incorporating reservoir computing into the original HAVOK algorithm. When compared to the original HAVOK algorithm, the RC-HAVOK can indeed model dynamical systems with a lower rank size, even if the error rates remain the

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

same. To clearly demonstrate this effect, we have compiled a table comparing the minimum/maximum/average error rates and their corresponding rank sizes for various dynamical systems using both the RC-HAVOK and the original HAVOK algorithms. The related table is depicted in Table XI.

VIII. CONCLUSION

In this research, a novel data-driven method for approximating the Koopman operator, the RC-HAVOK algorithm, is introduced. Reservoir Computing is not utilized in the conventional predictive manner. While RC typically demonstrates strong predictive capabilities, particularly for dynamical systems with straightforward readout mechanisms, the RC-HAVOK algorithm diverges from this approach. Instead of forecasting future variables or outcomes, the primary objective of this study is to derive a linear approximation of nonlinear dynamical systems. In this context, RC is leveraged to enhance the linear modeling aspect of the traditional HAVOK method. By integrating Reservoir Computing and the Hankel Alternative View of Koopman, the algorithm aims to reduce the size of the linear Koopman operator while achieving a lower error rate. The study evaluates the accuracy and feasibility of the RC-HAVOK algorithm across various dynamical systems featuring both smooth and non-smooth activation functions. The proposed method successfully applies to dynamic systems with smooth activation functions, including quadratic, cubic, and hyperbolic tangent. However, linear modeling could not be achieved with the desired accuracy in dynamic systems featuring non-smooth activation functions, such as the PWL function.

Implementations reveal that the proposed model outperforms other data-driven algorithms, especially when applied to widely used Lorenz time series data. However, a trade-off is observed between modeling time and rank size, where a decrease in one corresponds to an increase in the other. Furthermore, the performance of the algorithm is influenced by the reservoir structure; while increasing the number of neurons reduces the error rate, it simultaneously adds complexity to the model.

Overall, in the broader research context, the objective is to identify the optimal reservoir structure to develop well-approximated Koopman operator models. Additionally, further research will explore new approximation methods to derive linear models for dynamical systems with non-smooth activation functions. These developments contribute to advancing our understanding of the inherent unpredictability of dynamical systems.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

G. Yılmaz Bingöl: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Resources (equal); Software (equal); Supervision (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **O. A. Soysal:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Resources (equal); Software (equal); Supervision (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **E. Günay:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Resources (equal); Software (equal); Supervision (equal); Validation (equal); Visualization (equal); Writing – review & editing (equal).

DATA AVAILABILITY

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

APPENDIX: PARAMETERS OF THE RC-HAVOK APPROACH

The detailed parameters of the RC-HAVOK approach for diverse dynamical systems featuring various nonlinearities and Lorenz-like systems are comprehensively listed in Tables XII and XIII, respectively. The time span values and sizes of matrices U, S, V, and H are consistent across all systems, whereas the rank size r and sizes of dV, A, and B may vary.

TABLE XII. RC-HAVOK parameters of the dynamical systems with different nonlinearities used in this study.

Parameters	Cubic Lorenz	Cubic Chua	Tangent Chua	PWL Chua
U	100 × 100	100 × 100	100 × 100	100 × 100
S	100 × 100	100 × 100	100 × 100	100 × 100
V	199 900 × 100	199 900 × 100	199 900 × 100	199 900 × 100
H	100 × 199 900	100 × 199 900	100 × 199 900	100 × 199 900
r	4	4	4	6
dV	199 895 × 4	199 895 × 4	199 895 × 4	199 895 × 6
A	3 × 3	3 × 3	3 × 3	5 × 5
B	3 × 1	3 × 1	3 × 1	5 × 1

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

TABLE XIII. RC-HAVOK parameters of the Lorenz-like dynamical systems used in this study.

Parameters	Quadratic Lorenz	Liu	Liu-Ling
U	100 × 100	100 × 100	100 × 100
S	100 × 100	100 × 100	100 × 100
V	199 900 × 100	199 900 × 100	199 900 × 100
H	100 × 199 900	100 × 199 900	100 × 199 900
r	4	8	6
dV	199 895 × 4	199 895 × 8	199 895 × 6
A	3 × 3	7 × 7	5 × 5
B	3 × 1	7 × 1	5 × 1

REFERENCES

[1]W. Ren, X. Na, Y. Li, and M. Han, "Physics-informed hierarchical echo state network for predicting the dynamics of chaotic systems," Expert Syst. Appl. **228**, 120155 (2023).
[2]B. Ovilla-Martinez, E. Tlelo-Cuautle, A. M. Gonzalez-Zapata, L. G. de la Fraga, and I. Cruz-Vega, "Enhanced FPGA implementation of echo state networks for chaotic time series prediction," Integration **92**, 48–57 (2023).
[3]C. Marone, M. Cocco, A. Gualandi, D. Faranda, and G. Mengaldo, "Deterministic and stochastic chaos characterize laboratory earthquakes," Earth Planet Sci. Lett. **604**, 117995 (2023).
[4]G. Ghosh *et al.*, "Secure surveillance systems using partial-regeneration-based non-dominated optimization and 5D-chaotic map," Symmetry (Basel) **13**, 1447 (2021).
[5]Y. Liu and H. H. C. Iu, "Antimonotonicity, chaos and multidirectional scroll attractor in autonomous odes chaotic system," IEEE Access **8**, 77171–77178 (2020).
[6]B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," Proc. Natl. Acad. Sci. **17**, 315 (1931).
[7]I. Mezić, "Spectral properties of dynamical systems, model reduction and decompositions," Nonlinear Dyn. **41**, 309–325 (2005).
[8]S. Maćešić, N. Črnjarić Žic, and I. Mezić, "Koopman operator spectrum for random dynamical systems," J. Nonlinear Sci. **30**, 2007–2056 (2020).
[9]R. Mohr, M. Budišić and I. Mezić, "Applied Koopmanism," Chaos **22**, 047510 (2012).
[10]B. Chen *et al.*, "Data-driven Koopman model predictive control for optimal operation of high-speed trains," IEEE Access **9**, 82233–82248 (2021).
[11]C. S. Kallesøe, J. N. Jensen, L. H. Andersen, C. M. Jensen, M. C. Frederiksen, and R. Izadi-Zamanabadi, "HAVOK model predictive control for time-delay systems with applications to district heating," IFAC-PapersOnLine (22nd IFAC World Congress, 2022), Vol. 56, pp. 2238–2243.
[12]M. Yadegar, M. Bakhtiaridoust, F. N. Irani, and N. Meskin, "Data-driven sensor fault detection and isolation of nonlinear systems: Deep neural-network Koopman operator," ET Control Theory Appl. **17**, 123–132 (2022).
[13]X. Xu, X. Liu, Y. Xiao, X. Zhang, and J. Liu, "Deep neural networks with Koopman operators for modeling and control of autonomous vehicles," IEEE Trans. Intell. Veh. **8**, 153–146 (2023).
[14]L. J. Ratliff, E. Ling, L. Zheng, and S. Coogani, "Koopman operator applications in signalized traffic systems," IEEE Trans. Intell. Transp. Syst. **23**, 3214–3225 (2022).
[15]R. G. Moulder, E. Martynova and S. M. Boker, "HAVOK extensions for psychological time series forecasting," Multivariate Behav. Res. **57**, 171–172 (2022).
[16]J. Ding, C. Cheng and Y. Zhang, "A Koopman operator approach for machinery health monitoring and prediction with noisy and low-dimensional industrial time series," Neurocomputing **406**, 204–214 (2020).
[17]N. Cassamo and J. W. van Wingerden, "On the potential of reduced order models for wind farm control: A Koopman dynamic mode decomposition approach," Energies (Basel) **13**, 6513 (2020).
[18]X. Li, Y. Liu, Z. Ping, Z. Yin, and T. Yang, "Deep Koopman model predictive control for enhancing transient stability in power grids," Int. J. Robust Nonlinear Control **31**, 1964–1978 (2021).
[19]F. Lunkeit, A. Tantet, V. Lucarini, and H. A. Dijkstra, "Crisis of the chaotic attractor of a climate model: A transfer operator approach," Nonlinearity **31**, 2221–2251 (2018).
[20]P. Schlatter, H. Azizpour, H. Eivazi, L. Guastoni, and R. Vinuesa, "Recurrent neural networks and Koopman-based frameworks for temporal predictions in a low-order model of turbulence," Int. J. Heat Fluid Flow **90**, 108816 (2021).
[21]S. H. Son, A. Narasingam and J. S. I. Kwon, "Data-driven feedback stabilisation of nonlinear systems: Koopman-based model predictive control," Int. J. Control **96**, 770–781 (2023).
[22]A. Surana, "Koopman operator framework for time series modeling and analysis," J. Nonlinear Sci. **30**, 1973–2006 (2020).
[23]E. Bollt, "On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD," Chaos **31**, 013108 (2021).
[24]M. Gulina and A. Mauroy, "Two methods to approximate the Koopman operator with a reservoir computer," Chaos **31**, 023116 (2021).
[25]J. H. Tu, K. K. Chen and C. W. Rowley, "Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses," J. Nonlinear Sci. **22**, 887–915 (2012).
[26]S. Sosnowski, P. Bevanda and S. Hirche, "Koopman operator dynamical models: Learning, analysis and control," Annu. Rev. Control **52**, 197–212 (2021).
[27]A. T. Ihler, D. J. Alford-Lago, C. W. Curtis, and O. Issan, "Deep learning enhanced dynamic mode decomposition," Chaos **32**, 033116 (2022).
[28]I. G. Kevrekidis, M. O. Williams and C. W. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," J. Nonlinear Sci. **25**, 1307–1346 (2015).
[29]E. M. Bollt, Q. Li, F. Dietrich, and I. G. Kevrekidis, "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator," Chaos **27**, 103111 (2017).
[30]J. L. Proctor, S. L. Brunton and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," Proc. Natl. Acad Sci. U.S.A. **113**, 3932–3937 (2016).
[31]J. L. Proctor, S. L. Brunton and J. N. Kutz, "Sparse identification of nonlinear dynamics with control (SINDYc)," IFAC-PapersOnLine (10th IFAC Symposium on Nonlinear Control Systems NOLCOS, 2016), Vol. 49, pp. 710–715.
[32]M. Stender, C. Geier and N. Hoffmann, "Data-driven reduced order modeling for mechanical oscillators using Koopman approaches," Front Appl. Math. Stat. **9**, 1124602 (2023).
[33]J. N. Kutz, K. Kaheman and S. L. Brunton, "SINDy-PI: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics," Proc. R. Soc. A **476**, 20200279 (2020).
[34]J. L. Proctor, E. Kaiser, S. L. Brunton, B. W. Brunton, and J. N. Kutz, "Chaos as an intermittently forced linear system," Nat. Commun. **8**, 19 (2016).
[35]P. Goyal, A. Forootani and P. Benner, "A robust SINDy approach by combining neural networks and an integral form," arXiv preprint arXiv:2309.07193 (2023).
[36]G. Tanaka *et al.*, "Recent advances in physical reservoir computing: A review," Neural Netw. **115**, 100–123 (2019).
[37]L. Larger *et al.*, "Photonic information processing beyond turing: An optoelectronic implementation of reservoir computing," Opt. Express **20**, 3241 (2012).
[38]E. A. Viktorov, A. Locquet, D. S. Citrin, G. O. Danilenko, A. V. Kovalev, and D. Rontani, "Impact of filtering on photonic time-delay reservoir computing," Chaos **33**, 013116 (2023).
[39]J. Bueno, A. Argyris and I. Fischer, "Photonic machine learning implementation for signal recovery in optical communications," Sci. Rep. **8**, 8487 (2018).
[40]M. Small, T. Lymburn, S. D. Algar, and T. Jüngling, "Reservoir computing with swarms," Chaos **31**, 033121 (2021).
[41]T. L. Carroll and L. M. Pecora, "Network structure effects in reservoir computers," Chaos **29**, 083130 (2019).
[42]E. Martynova, R. G. Moulder and S. M. Boker, "Extracting nonlinear dynamics from psychological and behavioral time series through HAVOK analysis," Multivariate Behav. Res. **58**, 441–465 (2023).
[43]E. N. Lorenz, "Deterministic nonperiodic flow," J. Atmos Sci. **20**, 130–141 (1963).

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing

[44]L. Liu, C. Liu, T. Liu, and K. Liu, "A new chaotic attractor," Chaos Solitons Fractals **22**, 1031–1038 (2004).
[45]S. Yan-Chen, L. Ling and L. Chong-Xin, "Experimental confirmation of a new reversed butterfly-shaped attractor," Chin. Phys. Soc. **16**, 1897–1900 (2007).
[46]M. Zhang and Q. Han, "Dynamic analysis of an autonomous chaotic system with cubic nonlinearity," Optik (Stuttg) **27**, 4315–43199 (2016).
[47]T. Matsumoto, "The double scroll family," IEEE Trans. Circuits Syst. **33**, 1072–1118 (1986).
[48]R. Madan, *Chua's Circuit: A Paradigm fir Chaos* (World Scientific, 1993).
[49]Z. Zhang, Y. Huang, S. Cai, F. Yu, H. Shen, and S. Du, "A new multi-scroll Chua's circuit with composite hyperbolic tangent-cubic nonlinearity: Complex dynamics, hardware implementation and image encryption application," Integration **81**, 71–83 (2021).
[50]C. Wah Wu, A. Huang, L. Pivka, and M. Franz, "Chua's equation with cubic nonlinearity," Int. J. Bifurcation Chaos Appl. Sci. Eng. **6**, 2175–2222 (1996).
[51]J. Song, J. Wu, C. Zhao, J. Yang, J. Zhao, and H. Leng, "A hybrid method using HAVOK analysis and machine learning for predicting chaotic time series," Entropy **24**, 408 (2022).
[52]S. Ouala *et al.*, "Learning latent dynamics for partially observed chaotic systems," Chaos **30**, 103121 (2020).
[53]M. Li and L. Jiang, "Deep learning nonlinear multiscale dynamic problems using Koopman operator," J. Comput. Phys. **446**, 110660 (2021).
[54]J. Axås and G. Haller, "Model reduction for nonlinearizable dynamics via delay-embedded spectral submanifolds," Nonlinear Dyn. **111**, 22079–22099 (2023).
[55]J. N. Kutz, E. Kaiser and S. L. Brunton, "Data-driven discovery of Koopman eigenfunctions for control," Mach. Learn. Sci. Technol. **2**, 035023 (2021).

02 September 2024 10:14:31

Published under an exclusive license by AIP Publishing