IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 20, 2019, accepted November 5, 2019, date of publication November 8, 2019, date of current version November 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952365

Phase Space Reconstruction-Based Conceptor Network for Time Series Prediction

ZHENG XU[1], LING ZHONG[2], AND ANGUO ZHANG[3,4]
[1]School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[2]Alibaba Group, Chengdu 610000, China
[3]College of Physics and Information Engineering, Fuzhou University, Fuzhou 350108, China
[4]Research Institute of Ruijie, Ruijie Networks Company Ltd., Fuzhou 350002, China

Corresponding author: Anguo Zhang (anrial@live.cn)

ABSTRACT The Conceptor network is a newly proposed reservoir computing (RC) model, which outperforms traditional classifiers, which can fail to model new classes of data for a supervised learning task. However, the reservoir structure design for the Conceptor is single, involving just a traditional random network, which has strong coupling between nodes and limits computing ability. This study focused on the reservoir topology design problem, and we propose a complex network Conceptor-based phase space reconstruction of time series. Several dynamical systems were chosen to build complex networks using a phase space reconstruction algorithm. The experiment results obtained using a mix of two irrational-period sines showed that the proposed phase space reconstruction reservoir topologies with the appropriate values of threshold provide Conceptors with extra reconstruction precision. Among them, the phase space reconstruction reservoir-based Lorenz system shows the best performance. Further experiments also identified the appropriate values of threshold of the phase space reconstruction method required to obtain optimal performance. The precision showed a non-linear decline with increase in memory load, and the proposed Lorenz phase space reconstruction reservoir maintained its advantages under different memory loads.

INDEX TERMS Conceptor, reservoir computing, phase space reconstruction, time series prediction.

I. INTRODUCTION

Reservoir computing (RC), a type of recurrent neural network (RNN), has attracted increasing interest due to its advantages of high accuracy, fast learning, easy training and global convergence [1]. Typical RC models, such as echo state network (ESN) [2] and liquid state machine (LSM) [3], have been widely applied to time series related applications, such as time series prediction [4]–[8], pattern classification [9]–[12], and anomaly detection [13].

It has been demonstrated that the reservoir layer, which is generally initialized to be large-scale and fixed-weight, plays a key role in the performance of reservoir computing. In [14], the authors proposed that in addition to the connection weights, the structures of the reservoir also have a non-negligible influence on the computational accuracy of RC on time series problems. Complex neural networks have been shown to be an effective toolkit for providing insight into the intrinsic nature of time series data from a global perspective, and have also been shown to be powerful approaches to handling nonlinear time series problems. Complex networks with differently-designed topologies of reservoir structures have been proposed recently, such as a small-world topology [15], a critical topology [16], a cortex-like distribution [17] and a modular topology [18]. Zhang and Small [19] introduced a transformation method from a pseudoperiodic time series to complex networks, by representing each cycle of the pseudoperiodic series as a basic node. Yang and Yang [20] built complex networks from the correlation matrices of time series. Gao and Jin [21] proposed a reliable and effective method for constructing complex networks from time series based on phase space reconstruction.

Recently, a novel computing framework for RC, the Conceptor network, has been proposed by Jeager [22]. Conceptor networks can store multiple patterns of inputs into a reservoir layer, filter the corresponding states using a "Conceptor" module, and finally reconstruct the input signals to achieve high classification accuracy. Conceptor networks have the significant advantage that they are capable of dynamically loading new classes of data patterns, a difficult problem for most other conventional supervised training-based classifiers which can fail to recognize new classes of the input pattern.

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

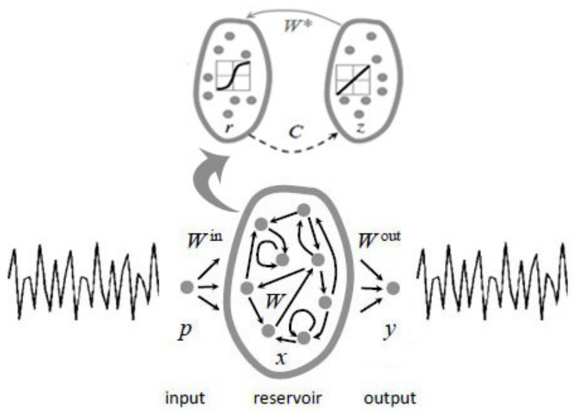

FIGURE 1. Conceptor network structure.

Because of these advantages, Conceptor networks have been applied to fields such as image classification [23], [24], natural language processing [25], time series classification [26].

In this paper, we propose a novel Conceptor network with an embedded reservoir of a time series phase space reconstruction complex network. Signal reconstruction of a mix of two irrational-period sines was employed as the network performance-testing platform. After analyzing the appropriate values of threshold of the phase space reconstruction method, an experiment comparing the network performance using this platform confirmed the advantages of the proposed reservoir over an ordinary random reservoir. The results demonstrate the effectiveness of the proposed reservoir network topology. The effects of memory load on network performance are also discussed.

The remainder of this paper are organized as follows: Sec. II briefly presents some basic preliminaries of the Conceptor network, and also describes the phase space reconstruction complex network based neural reservoir. Sec. IV examines the reconstruction of the mix of two irrational-period sines to analyze the performance of the Conceptor network with a modified reservoir structure. Sec. V presents the discussion and the conclusions.

II. BASIC PRINCPLES OF CONCEPTOR NETWORKS

Consider a recurrent neural network including an input layer of I neurons, an output layer of O neurons, and a hidden layer (dynamical reservoir) of H recurrently connected neurons. If we regard the reservoir as a two-layer network with the same number of neurons, as shown in Fig.1, then the reservoir state at time step $(n+1)$ can be expressed as

$$\mathbf{r}(n+1) = f\left(\mathbf{W}^{*}\mathbf{z}(n) + \mathbf{W}_{in}\mathbf{p}(n+1) + b\right) \quad (1)$$
$$\mathbf{z}(n+1) = \mathbf{C}(n)\mathbf{r}(n+1) \quad (2)$$

where $f(\cdot)$ is an activation function, $\mathbf{p} \in \mathbb{R}^{I}$, $\mathbf{x} \in \mathbb{R}^{H}$, $\mathbf{y} \in \mathbb{R}^{O}$ denote the network input, reservoir state and network output, respectively, and $\mathbf{r} \in \mathbb{R}^{H}$ and $\mathbf{z} \in \mathbb{R}^{H}$ are two recurrent states. Further, $\mathbf{W}_{in} \in \mathbb{R}^{H\times I}$ are the input weights, $\mathbf{W}_{out} \in \mathbb{R}^{O\times H}$ are the output weights, and $b \in \mathbb{R}^{H}$ is the network bias. The reservoir weights $\mathbf{W} \in \mathbb{R}^{H\times H}$ are resolved by two independent weights $\mathbf{W}^{*} \in \mathbb{R}^{H\times H}$ and $\mathbf{C}(n) \in \mathbb{R}^{H\times H}$, where $\mathbf{C}(n)$ is named as the Conceptor that characterizes the hidden activation of state $\mathbf{x}$.

It should be noted that similar to other reservoir computing frameworks like ESN or LSM, the values of the connection weights $\mathbf{W}_{in}$, $\mathbf{W}^{*}$, as well as the bias, b, are fixed once randomly initialized. Only the output weights, $\mathbf{W}_{out}$, need to be trained so that the network output $\mathbf{y}$ can reconstruct the corresponding input signal $\mathbf{p}$ as precisely as possible. According to Eq. 1 and Eq. 2, the network model can be rewritten as

$$\mathbf{x}(n+1) = \mathbf{C}(n)\cdot f\left(\mathbf{W}^{*}\mathbf{z}(n) + \mathbf{W}_{in}\mathbf{p}(n+1) + b\right) \quad (3)$$
$$\mathbf{y}(n+1) = \mathbf{W}_{out}\mathbf{x}(n) \quad (4)$$

The $\mathbf{W}_{out}$ is calculated by using the pseudo-inverse:

$$P = E[x(n)], \quad T = E[d(n)] \quad (5)$$
$$W_{out} = TP^{+} \quad (6)$$

where $\mathbf{d}(n)$ is the desired output at time step n.

Conceptor $\mathbf{C}$ can be calculated using a stochastic gradient descent method. The gradient of $\mathbf{E}[\| \mathbf{x} - \mathbf{Cx} \|^2] + \alpha^{-2} \| \mathbf{C} \|_{fro}^2$ with respect to $\mathbf{C}$ is

$$\frac{\partial}{\partial \mathbf{C}}\mathbf{E}[\| \mathbf{x} - \mathbf{Cx} \|^2] + \alpha^{-2} \| \mathbf{C} \|_{fro}^2 = \mathbf{I} - \mathbf{C}\mathbf{E}[\mathbf{x}\mathbf{x}^T] - \alpha^{-2}\mathbf{C} \quad (7)$$

where α is a design parameter that satisfies $\alpha \geq 0$. Thus, we can find an optimal solution to minimize the objective function by adjusting α. An online adaptive function of the Conceptor is defined as

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \lambda\left(\left(\mathbf{x}(n) - \mathbf{C}(n)\mathbf{x}(n)\right)\mathbf{x}^T - \alpha^{-2}\mathbf{C}(n)\right) \quad (8)$$

where λ denotes the learning rate.

We can load K input patterns $\mathbf{p}^j (j = 1, 2, \ldots, K)$ into a reservoir of dimension N. To reconstruct the different input patterns which feed into the reservoir, the Conceptor network needs to perform three steps: initialization, cueing and recalling:

(1) Initialization: The reservoir states begin with zero, and are driven by p^j with a length of $n_{washout}$.

(2) Cueing: The reservoir is driven by p^j with a length of n_{cue}; the corresponding $\mathbf{C}^j$ is initialized with a zero matrix, and is updated using Eq. 9.

$$r(n+1) = \tanh\left(Wr(n) + \mathbf{W}_{in}\mathbf{p}^j(n) + b\right)$$
$$\mathbf{C}^j(n+1) = \mathbf{C}^j(n) + \lambda_{cue}\left(\left(r(n) - \mathbf{C}^j(n)r(n)\right)\mathbf{r}^T(n) - \alpha^{-2}\mathbf{C}^j(n)\right) \quad (9)$$

(3) Recalling: The reservoir continues updating its internal state by using the input analog matrix $\mathbf{D}$ instead of using an

input signal, where the calculation method of **D** has been proposed [22]. $\mathbf{C}^{j}_{cue}$ autonomously self-updates according to Eq.(10) with a length of n_{recall}, and ultimately, the $\mathbf{C}^{j}_{recall}$ can be obtained.

$$
\begin{aligned}
z(n+1) &= \mathbf{C}^{j}(n) + \tanh\left(\mathbf{W}\mathbf{z}(n) + \mathbf{D}\mathbf{z}(n) + b\right) \\
\mathbf{C}^{j}(n+1) &= \mathbf{C}^{j}(n) + \lambda_{recall}\Big(\big(\mathbf{z}(n) \\
&\quad - \mathbf{C}^{j}(n)\mathbf{z}(n)\big)\mathbf{z}^{T}(n) - \alpha^{-2}\mathbf{C}^{j}(n)\Big) \qquad (10)
\end{aligned}
$$

III. PROPOSED METHOD

A. PHASE SPACE RECONSTRUCTION COMPLEX NETWORK GENERATION ALGORITHM

To generate a complex network of the reservoir for phase space reconstruction of a specific time series, we proposed a novel network generation algorithm which can be divided into two parts: phase space reconstruction of the time series, and construction of a complex network.

We can divide the time series into fixed length of sequence using the phase space reconstruction method, and map the sequences for the network nodes. The correlation coefficient between nodes determines whether there is a connection between nodes. When the correlation coefficient is greater than a certain threshold, these two nodes are linked together. According to Takens' embedding theorem [16], which is very often invoked as the motivation for applying a time delay embedding to reconstruct a phase space from a time series, the algorithm can be described as follows. For time series $\mathbf{z}(it)(i = 1, 2, \ldots, M)$, t is a sample interval and M is the sample length. Using the coordinate delay embedding algorithm we can calculate the phase space vector:

$$
\begin{aligned}
\overrightarrow{\mathbf{x}_k} &= \mathbf{x}_k(1), \mathbf{x}_k(2), \ldots, \mathbf{x}_k(m) \\
&= \mathbf{z}(kt), \mathbf{z}(kt+\tau), \ldots, \mathbf{z}\big(kt + (m-1)\tau\big) \qquad (11)
\end{aligned}
$$

where τ is the time delay, **m** is the embedding dimension, and $k = 1, 2, \ldots, N$. N is the number of total vectors in the reconstructed phase space, $N = M-(m-1)\tau/t$. In this work the time delay τ is determined by the $\mathbf{C}_C$ method, and using the false nearest neighbors (FNN) method to determine the time series' embedding dimension m.

To build a complex network corresponding to the time series, each vector point in the phase space can be seen as a node of the network, and whether there is a connection between nodes is determined by their distance in phase space. Given two vector points $\overrightarrow{\mathbf{X}_i}$ and $\overrightarrow{\mathbf{X}_j}$ $(i > j)$, their distance in phase space is defined as:

$$
\mathbf{d}_{ij} = \sum_{n=1}^{m} \| \mathbf{X}_i(n) - \mathbf{X}_j(n) \| \qquad (12)
$$

where $\mathbf{X}_i(n)$ is the nth elements of $\overrightarrow{\mathbf{X}_i}$, $\mathbf{X}_i = \mathbf{z}\big(i{*}t+(n-1)\tau\big)$. According to Eq.(10), we can obtain a distance matrix $\mathbf{D} = (\mathbf{d}_{ij})$ $(i > j)$.

We can choose a suitable threshold and establish the network using the following algorithm: there exists connection between node i and j if $|\ \mathbf{d}_{ij}\ | \leq \mathbf{r}_c$, and it is a directed connection (from i to j). However, if $|\ \mathbf{d}_{ij}\ | > \mathbf{r}_c$, the nodes are not connected. Consider the phase space distance between nodes as connection weights. We can then get a connection weight matrix $\mathbf{W} = (\mathbf{w}_{ij})$. $\mathbf{w}_{ij} = 0$ means there is no connection between nodes i and j . If $\mathbf{w}_{ij} \neq 0$, $\mathbf{w}_{ij} = \mathbf{d}_{ij}$. The connection weight matrix **W** determines the network's topological structure of time series phase space reconstruction.

B. MEMORY MANAGEMENT

The memory management mechanism of the network is as follows:

1. Gradually add the kinetic model into the concept machine network: If the models $p^1, p^2, \ldots, p^k$ are already stored in the reserve pool, a new model p^{k+1} can be stored without disturbing the previously stored models, even without knowing what model was previously-stored.

2. Measure the remaining memory capacity of the reserve pool; that is, the capacity of the reserve pool to store kinetic models.

3. Reduce redundancy: If a new model has some similarity to a stored model, loading it requires less memory than a new model that is completely different from the existing model.

The initial model storage procedure in Sec.II is to rewrite the initial random weight matrix $\mathbf{W}^*$ into a matrix **W** that stores the input model, and in the memory mechanism, this initial matrix can be more conveniently preserved. The value change is recorded into an input analog matrix **D** as shown in Eq.10:

$$
\begin{aligned}
X^{j}(n+1) &= \tanh\left(W^{*}x^{j}(n) + W_{in}p^{j}(n+1)\right) \\
&\approx \tanh\left(W^{*}x^{j}(n) + Dx^{j}(n)\right) \qquad (13)
\end{aligned}
$$

In a non-progressive training model, regularized linear regression can be used to calculate **D** to minimize the following variance formula:

$$
D = \operatorname{argmin}_{\tilde{D}} \sum_{j=1,\ldots,k} \sum_{n=1,\ldots,L} \| W_{in}p^{j}(n) - \tilde{D}x^{j}(n+1) \|^2
$$

where K is the number of loaded models and L is the length of the training samples. The reserve pool obtained in this way has a $\mathbf{W}^* + \mathbf{D}$ which is essentially the same as the weight matrix **W**.

The input simulation matrix $D_j(j = 1, \ldots, K)$ must satisfy the following conditions:

4. When the jth input simulation matrix D_j is combined with the concept machine C_i corresponding to the stored model $p^i(i \leq j)$, the autonomous power system $x(n+1) = C^i \tanh\left(W^{*}x^{j}(n) + D^{j}x^{j}(n) + b\right)$ can be heavy Construct the ith model.

5. To calculate D^{j+1}, you cannot use the stored models or their concept machines. Just use the new input model p^{j+1} to drive the network to get the necessary training data.

6. If the two training models are identical, i.e., $p^i = p^j(i > j)$, then $D^j = D^{j-1}$. The model p^i has been stored in the

network, and when the model reappears, the network does not need to change again to accommodate it. When p^i is similar to, but not equal to $p^j (i > j)$, the network can reduce its information redundancy. In this case, the increase in memory consumption will be lower.

The key to the memory management mechanism is to track the portion of the memory pool where the memory space is occupied by existing models, or the portion of the memory space that has not been developed. After storing the models $p^1, \ldots, p^j$, the occupied memory space can be described as $A^j = \vee(C^1, \ldots, C^j)$, and the development part can be represented as its complement $\neg A^j$.

Let the initial weight matrix of the N-dimensional reserve pool be W^*, and the training model sequence be $p^j(n) j = 1, \ldots, K; n = 1, \ldots, L$. The incremental storage algorithm steps are as follows:

Step 1: Initialize. $D^0 = A^0 = 0_{N*N}$. Select an "aperture" α for all models.

Step 2: Incrementally store the input model. The specific steps are as follows:

1) According to equation (2.9), when the input model p^j of length L drives the reserve pool, the state set $X^j = \{x^j(1), \ldots, x^j(L-1)\}, R^j = X^j(X^j)^T/(L-1)$ is obtained. Similarly, the input model set $P^j = \{p^j(2), \ldots, p^j(L)\}$ is obtained.

2) Calculate the corresponding concept machine $C^j = R^j(R^j + \alpha^{-2} I)^{-1}$.

3) Calculate the $N*N$ matrix D^j_{inc} (which will be added as an increment to D^{j-1}):

a) $F^{j-1} \neg A^{j-1}$, calculate the available memory space;

b) $T = W_{in} P^j - D^{j-1} X^j$, the matrix T is a target matrix in the linear regression of D^j_{inc};

c) $S = F^{j-1} X^j$, matrix S is a target matrix in linear regression;

d) $D^j_{inc} = \left(\left(\frac{SS^T}{L-1} + \alpha^{-2} I \right)^{+} \frac{ST^T}{L-1} \right)'$, linear regression equation

4) Update $D : D^j = D^{j-1} + D^j_{inc}$.

5) Update $A : A^j = A^{j-1} \vee C^j$.

IV. EXPERIMENTAL DESIGN AND RESULTS

A. PHASE SPACE RECONSTRUCTION COMPLEX NETWORK STRUCTURAL ANALYSIS

The purpose of generating a complex network is to establish a reservoir model with better dynamic characteristics for the Conceptor network. We compared several time series reconstruction complex networks to investigate their connection topologies and network characters.

a) Lorenz chaotic system:

$$\begin{aligned} \mathbf{d}_x/\mathbf{d}_t &= \sigma(y - x), \\ \mathbf{d}_y/\mathbf{d}_t &= x(\rho - z) - y, \\ \mathbf{d}_z/\mathbf{d}_t &= xy - \beta z, \end{aligned} \tag{14}$$

where $\sigma = 16$, $\rho = 45.92$, and $\beta = 4$.

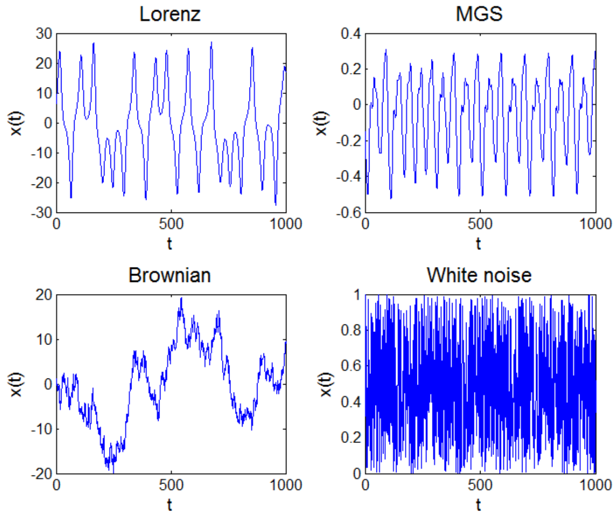

FIGURE 2. The time series form four kinds of dynamic models.

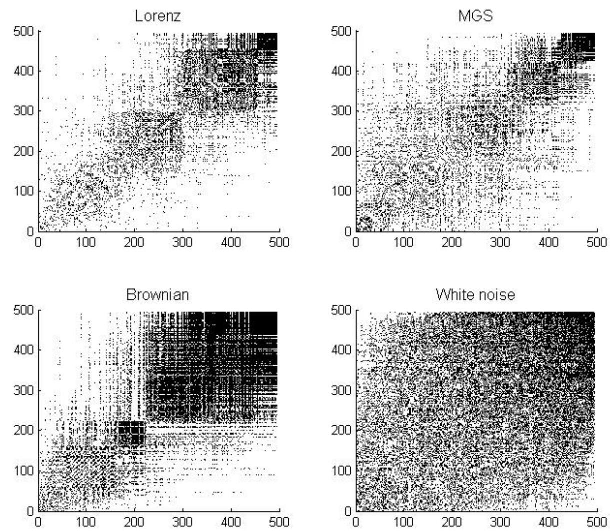

FIGURE 3. The complex network topologies from four time series according to phase-space reconstruction.

b) Mackey-Glass chaotic system:

$$y[n+1] = y[n] + \delta \left(\frac{0.2 y[n - \frac{\tau}{\delta}]}{1 + y[n - \frac{\tau}{\delta}]^{10}} - 0.1 y[n] \right), \tag{15}$$

where $\delta = 0.1$, $\tau = 17$ produces a mildly chaotic behavior.

c) Brownian motion system, representing a self-similar system model.

d) Gaussian white noise.

The time series of the four dynamic models above are described as shown in Fig. 2. Using the phase space reconstruction method, we can construct complex networks corresponding to these four dynamic systems. Fig. 3 shows the adjacency matrix of the complex networks as a scatter diagram. The adjacency matrix is ordered by ascending node degrees. Black points indicate that there is a connection between nodes whose number is equal to this point's x- and

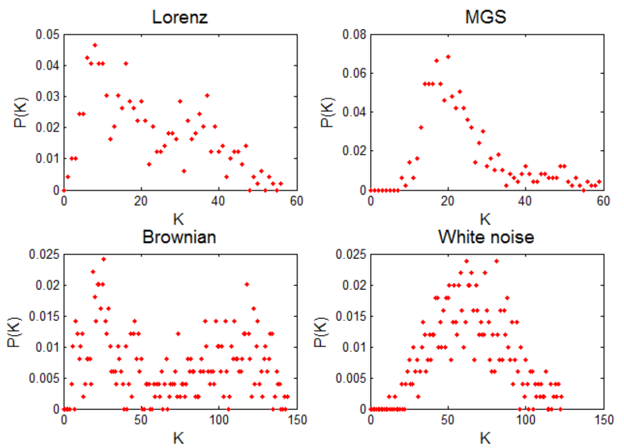

FIGURE 4. The network degree distributions from four time series according to phase-space reconstruction.

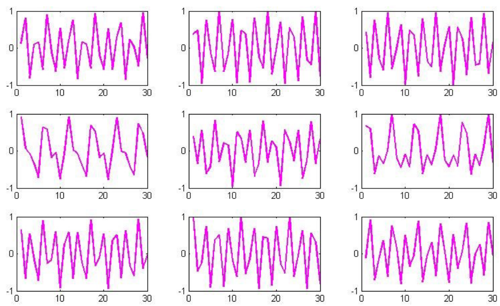

FIGURE 5. Mixed irrational periodic sine signal.

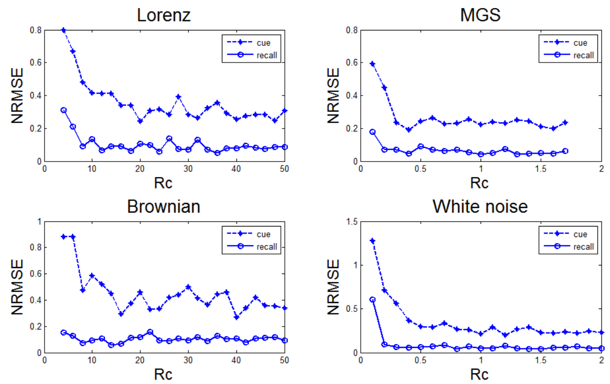

FIGURE 6. Effects of threshold RC on reconstruction accuracy for all reservoirs.

y- values. We can see that the complex networks generated from chaotic systems (Lorenz and MG) and self-affine systems (Brownian motion) have a clustered structure, while the nodes of the complex network representing the white noise series are not clustered. From the corresponding node degree distribution (Fig. 4), it is apparent that the degree distribution of the white noise network is Gaussian. The chaotic system network has a Weibull distribution. The self-affine systems have multiple peak distributions.

B. EXPERIMENTAL DESIGN

We used a mix of two irrational-period sines as input patterns to test the performance of the Conceptor networks. Two sines of period lengths $\sqrt{30}$ and $\sqrt{30}/2$ were added with random phase angles and random amplitudes, where however the two amplitudes were constrained to sum to 1. The different phase and amplitudes evolved different input patterns. Fig. 5 shows nine mixed irrational-period sines drawn from a 2-parametric family.

The Conceptors in the experiments had one input neuron with a mix of two irrational-period sine sequences. The reservoir was driven by a mix of 30 two irrational-period sines with different phase angles and amplitudes. The exact number of the reservoir neurons N, which were set to 200, is task-dependent. Tab. 1 presents the other parameters.

The prediction performance was measured using the normalized RMSE (NRMSE) and its standard deviation. The NRMSE is computed as follows:

$$NRMSE = \sqrt{\frac{\sum_{i=1} \left(\mathbf{l}_{test}\mathbf{y}_{test} - \mathbf{d}_{test}\right)^2}{\mathbf{l}_{test}\sigma^2}} \tag{16}$$

where $\mathbf{y}_{test}[n]$ is the network output during the testing phase; $\mathbf{d}_{test}[n]$ is the desired output during the testing phase and σ^2 is the variance of the desired output. The standard deviation of NRMSE (δ) is defined as follows: $NRMSE_{av} = \sum_{i=1}^{i=k} NRMSE(i)/k$. k denotes the number of independent trials.

C. ANALYSIS OF PREDICTION ACCURACY FOR ALL RESERVOIRS

The complex networks from different time series have different structural characteristics, and have different performance. In this section, we describe the results of our analysis of the performance of the conventional random reservoir and four phase space reconstruction reservoirs. For the reservoirs described as Sec. III-A, we first analyzed the effects of the threshold of phase space reconstruction method (RC) on reconstruction accuracy on the network's density and number of connections, factors which are significant for reservoir performance. Fig.6 shows the NRMSEs of four phase space reconstruction networks with different values of RC. We can see that the NRMSEs are stable when RC is larger than 15 for Lorenz system, 0.3 for MGS system, 12 for Brownian system and 0.5 for white noise. A comparison of the reconstruction performances between the traditional random reservoirs with the four phase space reconstruction reservoirs was performed.

The experiment compared the reconstruction capability of five kinds of reservoir structures, including a random network and the four phase space reconstruction networks. Load 30 mixed sine signal input model above on five reservoirs respectively. Fig. 7 shows reconstruction results of the first three input patterns from Conceptor $\mathbf{C}_{recall}$. The broken black represents the input training pattern and bold light pink is the reconstructed pattern. We can see that the random network

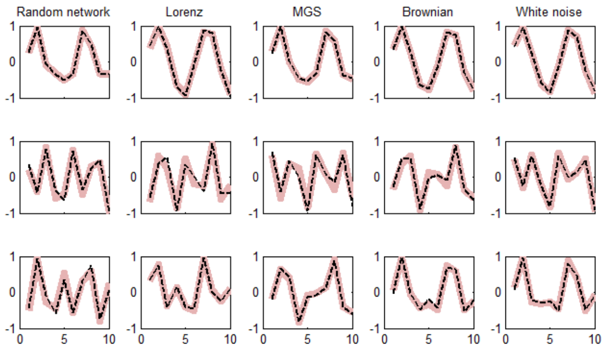

FIGURE 7. The reconstructed effect with Conceptor C^j_{recall}. Broken black is the original training pattern and bold light pink is the reconstructed pattern.

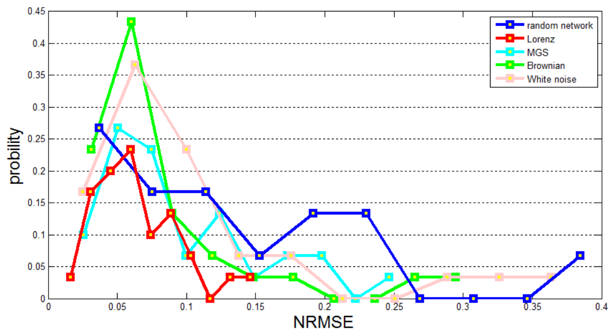

FIGURE 8. The reconstructed effect with Conceptor C^j_{recall}. Broken black is the original training pattern and bold light pink is the reconstructed pattern.

and four phase space reconstruction networks can reconstruct the input models effectively.

To compere the reconstruction ability of Conceptors $\mathbf{C}_{recall}$ with five reservoirs, we computed the reconstruction NRMSE of 30 patterns for each reservoir, and drew their NRMSE (Fig. 8). The blue indicates the NRMSE distribution of random networks, and the other four colors represent the four phase space reconstruction networks. The reconstruction precision of the four phase space reconstruction reservoirs is better than that of the random reservoir in general. Compared with a random network, their NRMSEs have smaller values. The phase space reconstruction network from the Lorenz system (red line) shows the best reconstruction precision. Tab. 2 shows the NRMSE mean and standard deviation δ for five kinds of reservoirs. The comparisons show that four phase space reconstruction networks have smaller average NRMSEs than the random network. Among them, the Lorenz system phase space reconstruction network shows the best precision (including average NRMSE and the corresponding δ), indicating that the proposed phase space reconstruction reservoir is the most powerful of the network models for reconstructing these patterns. We then observed the NRMSEs of the phase space reconstruction of the Lorenz system network for each pattern using $\mathbf{C}_{cue}$ and $\mathbf{C}_{recall}$. As shown in Fig. 9, the reconstruction ability of the Concepor network is improved from $\mathbf{C}_{recall}$ to $\mathbf{C}_{cue}$, due to the autonomous update of the Conceptor at the recall stage.

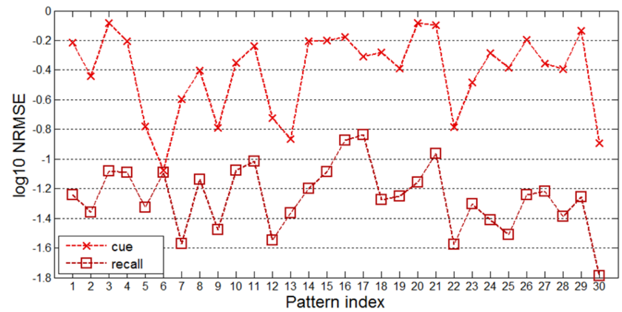

FIGURE 9. The reconstruction NRMSE with C^j_{cue} and C^j_{recall} of phase-space reconstruction network model.

D. ANALYSIS OF MEMORY LOAD EFFECTS ON RECALL ACCURACY

The calculation accuracy reflects the calculating ability of the Conceptor networks and is affected by memory load of reservoir. To explore the relationship between recall accuracy and memory load, and analyze the reservoir's memory capacity for dynamic models, we carried out a further experiment. We gradually increased the number of input models and observed the changes in the reconstruction error depending upon the memory loads, as follows:

1) Construct a reservoir network structure.

2) In separate trials, load this reservoir with an increasing number k of patterns (ranging from $k = 2$ to $k = 100$ for the mixed sines). Since the change in accuracy is obvious when the input model number is small and the error tends to be stable because the memory space is close to saturation when the number is great, we set $K = [2, 3, 5, 8, 12, 16, 25, 50, 100]$, instead of using a uniform distribution.

TABLE 1. Parameter settings in the experiments.

Parameter	Value	Parameter	Value
K	20	N	200
α	100	$n_{washout}$	100
n_{cue}	30	n_{recall}	10000
λ_{cue}	0.01	λ_{recall}	0.01

3) After loading, repeat the recall scheme described above, with the parameters as shown in Table 1. Monitor the recall accuracy obtained from $\mathbf{C}^j_{recall}$ for the first 10 of the loaded patterns (if less than 10 were loaded, do it only for these).

4) Repeat step 2) and 3) five times, with freshly created patterns, using the same reservoir.

The results of the comparison between the phase space reconstruction complex networks and random networks are shown in Fig. 8. The plotted curves are the averages over 10 recall targets and the five experiment repetitions. Each point in the diagrams thus reflects an average over 50 NRMSE values, except in cases where $K < 10$ patterns were stored; then plotted values correspond to averages over $5 * K$ NRMSE values for recalling of loaded patterns. The error bar of each point reflects the standard deviation. The main findings from Fig. 8 are listed as follows:

TABLE 2. Average NRMSE and standard deviations for five kinds of reservoirs.

	Average NRMSE$\times 10^{-2}$	Standard Deviation $\times 10^{-2}$
Loreza	6.22	3.09
MGS	8.28	5.75
White Noise	7.70	8.12
Brownian	9.38	7.88
Cortex Network	11.03	7.76
Random	13.01	9.34

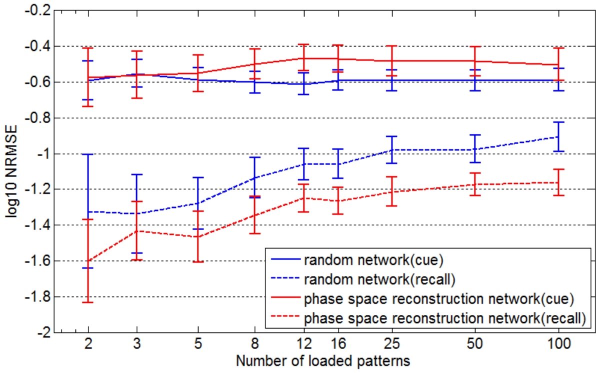

FIGURE 10. 10 Effects of memory load on recall accuracy.

1) For all numbers of stored patterns and all reservoir models, the reconstruction NRMSEs from $\mathbf{C}_{recall}$ are smaller than NRMSEs from $\mathbf{C}_{cue}$.

2) For all reservoir models, the reconstruction NRMSEs from $\mathbf{C}_{recall}$ show a non-linear increase with increase in memory load.

3) Although the NRMSEs of random network from $\mathbf{C}_{cue}$ are better than those of the phase space reconstruction when the memory load is larger than four, the phase space reconstruction network's accuracy from $\mathbf{C}_{recall}$ is greater than those of the random networks for different memory loads. The accuracy from $\mathbf{C}_{recall}$ is the most accurate indicator of evaluation because $\mathbf{C}_{recall}$ is more mature than $\mathbf{C}_{cue}$. These results indicate that the phase space reconstruction networks outperform the traditional random networks.

V. CONCLUSION

This study describes a phase space reconstruction reservoir generation algorithm for the Conceptor network, which was investigated using a test involving the reconstruction of a mix of two irrational-period sines. The simulation results demonstrate the effectiveness of the proposed reservoir network topology. The precision of phase space reconstruction reservoirs improves with the increase in threshold, because of the increase in reservoir network density. Four phase space reconstruction reservoirs had improved computational capability over the conventional random reservoir, and the improvement in the reconstruction precision of the Lorenz phase space reconstruction reservoir and its standard deviation was largest. Further experiments revealed that reconstruction precision showed a non-linear decline with increase in memory load, and the proposed Lorenz phase space reconstruction reservoir maintained its advantages for different amounts of memory load. The main advantage of the proposed reservoir topology is to generate an efficient complex network reservoir which can improve the computing abilities of Conceptor networks.

REFERENCES

[1] D. Hai-Yan, P. Wen-Jiang, and H. Zhen-Ya, "A multiple objective optimization based echo state network tree and application to intrusion detection," in *Proc. IEEE Int. Workshop VLSI Design Video Technol.*, May 2005, pp. 443–446.

[2] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, Apr. 2004.

[3] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, 2002.

[4] Z. Shi and M. Han, "Support vector echo-state machine for chaotic time-series prediction," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 359–372, Mar. 2007.

[5] P. Yu, W. Jian-Min, and P. Xi-Yuan, "Traffic prediction with reservoir computing for mobile networks," in *Proc. 5th Int. Conf. Natural Comput.*, Aug. 2009, pp. 464–468.

[6] S. L. Goh, M. Chen, D. H. Popović, K. Aihara, D. Obradovic, and D. P. Mandic, "Complex-valued forecasting of wind profile," *Renew. Energy*, vol. 31, no. 11, pp. 1733–1750, 2006.

[7] Q. Ma, Q. Zheng, H. Peng, and J. Qin, "Chaotic time series prediction based on fuzzy boundary modular neural networks," *Acta Phys. Sinca*, vol. 58, no. 3, p. 1410, 2009.

[8] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Netw.*, vol. 20, no. 3, pp. 335–352, 2007.

[9] M. D. Skowronski and J. G. Harris, "Automatic speech recognition using a predictive echo state network classifier," *Neural Netw.*, vol. 20, no. 3, pp. 414–423, Apr. 2007.

[10] M. J. Embrechts, L. A. Alexandre, and J. D. Linton, "Reservoir computing for static pattern recognition," in *Proc. Eur. Symp. Artif. Neural Netw.*, 2009, pp. 245–250.

[11] L. A. Alexandre, M. J. Embrechts, and J. Linton, "Benchmarking reservoir computing on time-independent classification tasks," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 2009, pp. 89–93.

[12] M. D. Skowronski and J. G. Harris, "Minimum mean squared error time series classification using an echo state network prediction model," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 3153–3156.

[13] Q. Chen, A. Zhang, T. Huang, Q. He, and Y. Song, "Imbalanced dataset-based echo state networks for anomaly detection," *Neural Comput. Appl.*, Oct. 2018, doi: 10.1007/s00521-018-3747-z.

[14] M. Maltz, *Psycho-Cybernetics*. Cambridge, MA, USA: MIT Press, 1965.

[15] Z. Deng and Y. Zhang, "Collective behavior of a small-world recurrent neural system with scale-free distribution," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1364–1375, Sep. 2007.

[16] M. Kaiser and C. C. Hilgetag, "Development of multi-cluster cortical networks by time windows for spatial growth," *Neurocomputing*, vol. 70, pp. 1829–1832, Jun. 2007.

[17] X. Li, L. Zhong, F. Xue, and A. Zhang, "A priori data-driven multi-clustered reservoir generation algorithm for echo state network," *PLoS ONE*, vol. 10, no. 4, 2015, Art. no. e0120750.

[18] Q.-L. Ma and W.-B. Chen, "Modular state space of echo state network," *Neurocomputing*, vol. 122, pp. 406–417, Dec. 2013.

[19] J. Zhang and M. Small, "Complex network from pseudoperiodic time series: Topology versus dynamics," *Phys. Rev. Lett.*, vol. 96, Jun. 2006, Art. no. 238701, doi: 10.1103/PhysRevLett.96.238701.

[20] Y. Yang and H. Yang, "Complex network-based time series analysis," *Phys. A, Stat. Mech. Appl.*, vol. 387, nos. 5–6, pp. 1381–1386, Feb. 2008, doi: 10.1016/j.physa.2007.10.055.

[21] Z. Gao and N. Jin, "Complex network from time series based on phase space reconstruction," *Chaos*, vol. 19, no. 3, 2009, Art. no. 033137.

[22] H. Jaeger, "Controlling recurrent neural networks by conceptors," 2014, *arXiv:1403.3369*. [Online]. Available: https://arxiv.org/abs/1403.3369

[23] Y. Hu, M. S. Ishwarya, and L. C. Kiong, "Classify images with conceptor network," 2015, *arXiv:1506.00815*. [Online]. Available: https://arxiv.org/abs/1506.00815

[24] G. Qian and L. Zhang, "A simple feedforward convolutional conceptor neural network for classification," *Appl. Soft Comput.*, vol. 70, pp. 1034–1041, 2018, doi: 10.1016/j.asoc.2017.08.016.

[25] T. Liu, J. Sedoc, and L. Ungar, "Correcting the common discourse bias in linear representation of sentences using conceptors," in *Proc. BioCreative/OHNLP Challenge*, 2018, pp. 256–260. [Online]. Available: https://arxiv.org/abs/1811.11002%0Ahttps://sites.google.com/view/ohnlp2018

[26] L. Wang, Z. Wang, and S. Liu, "An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm," *Expert Syst. Appl.*, vol. 43, pp. 237–249, Jan. 2016, doi: 10.1016/j.eswa.2015.08.055.

ZHENG XU received the master's degree from the Gelsenkirchen University of Applied Sciences, Germany. He is currently pursuing the Ph.D. degree in mechanical engineering with the Nanjing University of Science and Technology, China. His research interests include artificial neural network and control theory and application.

LING ZHONG received the B.S. and M.S. degrees in control engineering from Chongqing University, Chongqing, in 2013 and 2016, respectively.

She has been an Algorithm Research and Development with Alibaba Group. Her research interests include machine learning, artificial neural networks, and recommend systems.

ANGUO ZHANG was born in Hefei, Anhui, China, in 1990. He received the bachelor's and master's degrees in control engineering from Chongqing University, Chongqing, in 2012 and 2016, respectively. He is currently pursuing the Ph.D. degree in communication and information systems with Fuzhou University. Since 2018, he has been a Researcher and a Senior Engineer with the Research Institute of Ruijie, Ruijie Networks Company Ltd. His research interests include machine learning, artificial neural networks, and control theory and applications.