Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Modeling deterministic echo state network with loop reservoir*

Xiao-chuan SUN†[1,2], Hong-yan CUI[1], Ren-ping LIU[2], Jian-ya CHEN[1], Yun-jie LIU[1]

([1]*Beijing Key Laboratory of Network System Architecture and Convergence,*
Beijing University of Posts and Telecommunications, Beijing 100876, China)

([2]*ICT Centre, Commonwealth Scientific and Industrial Research Organization, Sydney 1710, Australia*)

†E-mail: jrsxc@163.com

Received Mar. 15, 2012; Revision accepted July 3, 2012; Crosschecked July 6, 2012

Abstract: Echo state network (ESN), which efficiently models nonlinear dynamic systems, has been proposed as a special form of recurrent neural network. However, most of the proposed ESNs consist of complex reservoir structures, leading to excessive computational cost. Recently, minimum complexity ESNs were proposed and proved to exhibit high performance and low computational cost. In this paper, we propose a simple deterministic ESN with a loop reservoir, i.e., an ESN with an adjacent-feedback loop reservoir. The novel reservoir is constructed by introducing regular adjacent feedback based on the simplest loop reservoir. Only a single free parameter is tuned, which considerably simplifies the ESN construction. The combination of a simplified reservoir and fewer free parameters provides superior prediction performance. In the benchmark datasets and real-world tasks, our scheme obtains higher prediction accuracy with relatively low complexity, compared to the classic ESN and the minimum complexity ESN. Furthermore, we prove that all the linear ESNs with the simplest loop reservoir possess the same memory capacity, arbitrarily converging to the optimal value.

Key words: Echo state networks, Loop reservoir structure, Memory capacity
doi:10.1631/jzus.C1200069 **Document code:** A **CLC number:** TP183

1 Introduction

Recurrent neural networks (RNNs) have been designed for modeling nonlinear dynamical systems (Siegelmann and Sontag, 1991; Mandic and Chambers, 2001; Tino *et al*., 2001; Abbasi Nozari *et al*., 2012). However, their complicated training process (Hochreiter *et al*., 2001) based on direct weight optimization, in addition to fading memory, has limited its application in practical engineering problems. To overcome training difficulty and the limit of fading memory, Jaeger (2001; 2002a) proposed a novel framework for RNNs, called the echo state network (ESN). The kernel component of ESN is a dynamic reservoir with abundant neurons that are randomly inter- and/or self-connected. The reservoir can transform input signals into a high-dimensional activation pattern, mitigating the fading memory of the input history effectively, while the optimal output weight is computed through a simple linear regression method, reducing the computation complexity. As a result of these merits, ESNs have attracted wide attention of researchers, and have been successfully applied in various research fields, including wireless communication channel equalization (Jaeger and Hass, 2004), dynamic pattern recognition (Ozturk and Principe, 2007), robot control (Salmen and Ploger, 2005), and especially time-series prediction (Deng and Zhang, 2007; Shi and Han, 2007; Xue *et al*., 2007; Holzmann and Hauser, 2010; Chatzis and Demiris, 2011; Xia *et al*., 2011).

So far, various extended ESN models have been explored, including a recurrent neural system with small-world and scale-free characteristics (Deng and Zhang, 2007), decoupled ESNs with lateral inhibition (Xue *et al*., 2007), support vector ESN combined with

* Project supported by the National Basic Research Program (973) of China (No. 2012CB315805), the Fundamental Research Funds for the Central Universities, China (No. 2009RC0124), the National Key Science and Technology Projects, China (No. 2010ZX03004-002-02), and the Australian Centre for Broadband Innovation (ACBI)

support vector machine (Shi and Han, 2007), ESNs with filter neurons and a delay & sum readout (Holzmann and Hauser, 2010), a complex ESN with full second-order statistical information (Xia *et al.*, 2011), and the novel ESN modeled using a Bayesian approach (Chatzis and Demiris, 2011). Although these schemes improve performance over the classic ESN in some respects, they also reflect some main problems. Some parameters, including the spectral radius, input/reservoir size, input/reservoir connection, and reservoir sparsity, need to be tuned individually in a trial-and-error way to satisfy the performance requirements of ESNs. Additionally, the reservoirs randomly generated are able to engender indeterministic weight matrices. Such uncertain reservoir structures make it difficult to analyze memory capacities (MCs) of the ESNs in theory.

To deal with the above problems, Rodan and Tino (2011) proposed three representative minimum complexity ESNs. Moreover, they proved the superior MC of the ESN with a simple cycle reservoir (SCR) and derived the closed-form formula for the MC. Using the simple reservoir structure and two free parameters, this SCR obtained only performance comparable to the classic ESN. It is not yet clear whether such simple deterministic ESNs, generated based on the simplest loop reservoir (i.e., SCR), are superior to the classic ESN, and whether they possess larger MCs. Besides, it is worth studying how to further reduce the number of free parameters. Hence, our motivation is to design a relatively simple reservoir structure, reduce the number of free parameters, and provide a good memory capacity while achieving a more competitive performance.

In this paper, we propose a novel simple ESN with loop reservoir structure, known as a deterministic ESN with an adjacent-feedback loop reservoir (ALR). The main contributions of this paper can be summarized as follows. First, a novel deterministic ESN is designed for time-series prediction. We introduce regular feedbacks to intensify the interactions between reservoir units and increase the reservoir sparsity, which effectively improves prediction performance (e.g., by about 2%, up to 6% for the NARMA task). Furthermore, we analyze the essential characteristics of the deterministic reservoir. Second, in ALR, only one free parameter r is tuned, leading to a successful compromise between complexity and prediction accuracy. In contrast to the situation in which two free parameters r and v are used (Rodan and Tino, 2011), the trial-and-error selection of free parameters is simplified while maintaining satisfactory performance. Finally, we prove that the ALR, combined with all the other deterministic linear ESNs evolving from the SCR, possesses the same superior MC as the SCR. It shows the advantages of the loop reservoir structure in terms of MC.

In the ALR reservoir, the feedback between adjacent units is introduced based on the SCR. Additionally, we consider the ESNs with i (i=2, 3, ···, N, N is the reservoir size) unit-feedback loop reservoirs. These schemes exhibit similar performance to the ALR, due to their similar symmetrical reservoir structures with the same complexity. Hence, in this paper, we focus on modeling the representative ALR. To reduce the free parameters, consider that the reservoir and input weight matrices have the same weight value $r\in(0, 1)$, and that the signs of input weights are generated by different methods (e.g., random, chaos). Moreover, the MC of the reservoirs represents their ability to recover information of the former inputs for a certain time (Jaeger, 2002b). We derive the general MC formula of all deterministic ESNs based on the SCR.

2 Echo state networks

The ESN, which is trained through supervised learning, is a special form of recurrent neural network. As shown in Fig. 1, the input signal $\boldsymbol{u}(t)$ drives the randomly generated reservoir, and then the output $\boldsymbol{y}(t)$ is evaluated by means of linear readout, using the updated reservoir state $\boldsymbol{x}(t)$. The ESN consists of K input units $\boldsymbol{u}(t)=(u_1(t), u_2(t), \cdots, u_K(t))^{\mathrm{T}}$, N reservoir units $\boldsymbol{x}(t)=(x_1(t), x_2(t), \cdots, x_N(t))^{\mathrm{T}}$, and L output units $\boldsymbol{y}(t)=(y_1(t), y_2(t), \cdots, y_L(t))^{\mathrm{T}}$. The real-valued connection weights are obtained from the $N\times K$ weight matrix $\boldsymbol{W}^{\mathrm{in}}$ for the input weight, from the $N\times N$ weight matrix $\boldsymbol{W}$ for the reservoir connections, from the $L\times N$ weight matrix $\boldsymbol{W}^{\mathrm{out}}$ for the connections from reservoir units to the output units. Each neuron or unit of ESN has an activation state at a given time step t. The activation states of reservoir units are generally updated using the following equation:

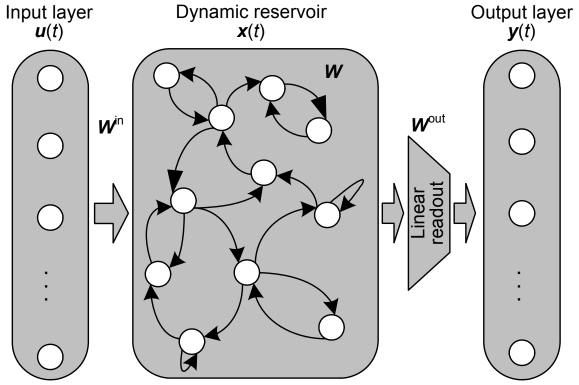

Fig. 1 Classic echo state network structure

$$x(t+1)=f(W^{\text{in}}u(t+1)+Wx(t)), \quad (1)$$

where f is the transfer function of the reservoir units (usually the hyperbolic tangent or other sigmoidal function, and maybe the identity function). The generic outputs are calculated according to the activation states by

$$y(t+1)=f(W^{\text{out}}x(t+1)). \quad (2)$$

In our case, the output function is usually linear.

In the standard ESN implementation, only the output matrix W^{out} is computed, and W^{in} and W keep unchanged after initialization. The spectral radius of the reservoir weight matrix is set to less than 1 to obtain the echo state property. To evaluate ESN performance, the normalized root mean square error (NRMSE) is used to measure the prediction accuracy, which is calculated by

$$\text{NRMSE}=\sqrt{\frac{1}{l_{\text{test}}\sigma^2}\sum_{i=1}^{l_{\text{test}}}\left(y_{\text{test}}(i)-d(i)\right)^2}, \quad (3)$$

where l_{test} is the number of test samples, $y_{\text{test}}(i)$ and $d(i)$ are the test output and desired output during the testing phase respectively, and σ^2 is the variance of the desired output.

3 Deterministic echo state network with a loop reservoir

To simplify the ESN, we propose a simple deterministic ESN with an adjacent-feedback loop reservoir (ALR). Unlike the reservoirs of previous ESNs, the novel reservoir is constructed by adding regular adjacent-feedbacks between adjacent reservoir units based on the simplest loop reservoir structure (i.e., SCR). Embedded in the ESN, the ALR possesses a collection of prominent features, including novel loop reservoir structure, only one tuned free parameter (r), and good memory capacity.

3.1 ALR

The system structure of the ALR is shown in Fig. 2. The input signal $u(t)$ impels the adjacent-feedback loop reservoir, which is used as input of linear readout to reconstruct the desired output $y(t)$. Unlike the randomly generated reservoir of the classic ESN, the ALR has a deterministic reservoir structure, in which the reservoir units are connected orderly in the loop manner, and each unit is also connected to the preceding one by adjacent feedback. The weight matrix W of the adjacent-feedback loop reservoir has the following symmetrical characteristics: for i=1, 2, ···, N, sub-diagonal $W_{i,i+1}=r$ and $W_{i+1,i}=r$, upper-right corner element $W_{1,N}=r$, and lower-left corner element $W_{N,1}=r$, where $r\in(0, 1)$ is the nonzero weight for all the connections. For example, with reservoir size N=5, the symmetric reservoir weight matrix W is mathematically formulated as

$$W=\begin{bmatrix}0&r&0&0&r\\r&0&r&0&0\\0&r&0&r&0\\0&0&r&0&r\\r&0&0&r&0\end{bmatrix}_{5\times5}.$$

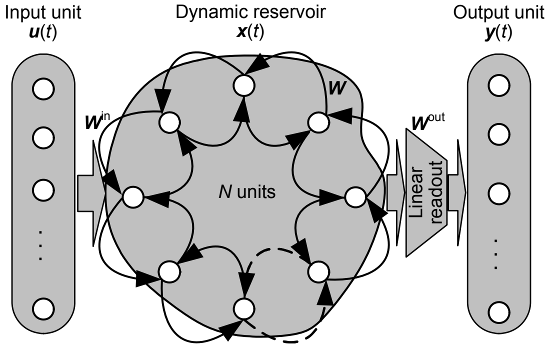

Fig. 2 System structure of the adjacent-feedback loop reservoir

The ESN training process (Lukosevicius and Jaeger, 2009) is a simple linear regression task. Consider the following training steps that ensure the prediction performance of the ALR:

1. Generate deterministic input weight matrix $\boldsymbol{W}^{\text{in}}$ and reservoir weight matrix $\boldsymbol{W}$. The input weight signs are generated by the random method or logistic map. To obtain the echo state property, set the spectral radius of the reservoir weight matrix $\boldsymbol{W}$ to less than 1.

2. Collect the network states $\boldsymbol{x}(t)$ into a row-wise matrix $\boldsymbol{X}$ after a washout time T_0, and use the ordered input and output sequences (u_{train}, y_{train}) to train the ALR.

3. Calculate the optimal output weights according to ridge regression (Wyffels *et al.*, 2008) by

$$\boldsymbol{W}^{\text{out}}=(\boldsymbol{X}^{\text{T}}\boldsymbol{X}+\lambda^2\boldsymbol{I})^{-1}\boldsymbol{X}^{\text{T}}\boldsymbol{y}, \tag{4}$$

where λ is the penalty factor for decreasing the effects of noise and overfitting and $\boldsymbol{y}$ is a vector of the output target values.

3.2 Model analysis

In fact, why the reservoir of ESN (Jaeger, 2001; 2002a; 2002b; 2002c) has such astonishing performance in approximating nonlinear systems is still not well understood. Many extensions and reconstructions of the reservoirs (Deng and Zhang, 2007; Xue *et al.*, 2007) can display the improved performance at the expense of inexplicable dynamic characteristics of the reservoirs. Besides, with respect to prediction performance, the existing SCR did not outperform the classic ESN, and was only comparable to it. Hence, we aim to model as simple a deterministic reservoir structure as possible while obtaining superior prediction performance. Based on the SCR, the ALR introduces adjacent-feedback connections between any adjacent reservoir units, which increases reservoir sparsity to some extent. As is well known, the reservoir sparsity can be used to measure the number of connections in reservoir structure, which directly affects the nonlinear approximation ability of ESNs. In other words, the more the connections, the stronger the ability to approximate nonlinear systems. Therefore, it is theoretically expounded that the ALR exhibits better performance than the SCR.

To decrease the number of tuned free parameters, we consider that the input absolute weights have the same value r as the reservoir ones, where r is drawn from a standard normal distribution (Zhang *et al.*, 2012) over (0, 1). Obviously, there exists the only free parameter r to be tuned, which considerably simplifies the ESN. Theoretically, although having only one free tuned parameter narrows the entire range of parameter variation, potentially leading to certain performance degradation, this can significantly simplify the trial-and-error process of selecting the two free parameters (Rodan and Tino, 2011) and greatly enhance the flexibility of the ALR. Therefore, using only one free parameter is able to maintain balance between complexity and prediction performance. Besides, input units are fully connected to reservoir units, and input weight signs are generated by means of the random method or logistic map (Rodan and Tino, 2011).

Jaeger (2002b) has investigated the short-term MC of the ESN, which is defined as the ability of the network to recover the values of former inputs. When the input stream $\cdots u(t-1)u(t)$ is fed into the ESN, for a given delay k, the k-delay MC of the ESN is expressed as

$$\text{MC}_k=\frac{\text{Cov}^2(\boldsymbol{u}(t-k),\boldsymbol{y}(t))}{\text{Var}(\boldsymbol{u}(t))\text{Var}(\boldsymbol{y}(t))}, \tag{5}$$

where $\boldsymbol{y}(t)$ is the predicted output, and Cov and Var denote the covariance and variance, respectively. The whole MC of the ESN is given by

$$\text{MC}=\sum_{k=1}^{\infty}\text{MC}_k. \tag{6}$$

Furthermore, Jaeger (2002b) demonstrated that if input/output units are independent and identically distributed (i.i.d.), the ESN's MC is not over N. With the same assumption, Rodan and Tino (2011) proved that the MC of the linear SCR network is

$$\text{MC}=N-1+r^{2N}. \tag{7}$$

Likewise, for the case of an i.i.d. zero-mean input stream, we theoretically derive the MC of the ALR, using the linear reservoir (i.e., identity reservoir activation). We find that the ALR has the same MC as the linear SCR network. More importantly, we further prove that all the linear ESNs with the loop reservoir structure (i.e., all the structure changes evolving from the SCR) possess the same MC as the linear SCR network. Assume that the input stream is a univariate time-series, and the input weight matrix is denoted by N-dimensional vectors, $\boldsymbol{W}^{\text{in}}=(W_1, W_2, \cdots, W_N)$. We propose a unit feedback vector $\boldsymbol{H}$ that denotes the delay variation of the reservoir units, i.e., $\boldsymbol{H}_k=(W_k, S_{k,1}, S_{k,2}, \cdots, S_{k,N-1})$, k=1, 2, ⋯, N, where $S_{k,j}$ (j=1, 2, ⋯, N−1) is defined as the sum of the input weights of all the units with j unit delays from the kth reservoir unit. Additionally, a regular matrix $\boldsymbol{\Psi}$ is denoted by $\boldsymbol{\Psi}=(\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_N)$.

Lemma 1 If the reservoir weight matrix has the same weight r, and $\boldsymbol{\Psi}$ is a regular matrix for input vectors, all the linear ESNs with the simplest loop reservoir structure have the same MC as the SCR. That is, their MC can be obtained from Eq. (7).

The proof can be found in the Appendix. Note that in the case of i.i.d. input stream, the MCs of these schemes can obviously converge to theoretical limitation N. Hence, it is theoretically proved that the ALR possesses the optimal MC. But for non-i.i.d. sources, the long- and short-range dependences between the input elements can improve the MC over the reservoir size N (Jaeger, 2002b).

4 Experimental results and performance evaluations

We conducted a comprehensive experimental evaluation of the ALR scheme, considering three scenarios: a NARMA system benchmark task, a nonlinear chaotic system, and real-world prediction tasks. To demonstrate the preponderances of our scheme, we also evaluated the classic ESN (CESN) and the SCR proposed by Rodan and Tino (2011). Moreover, we analyzed where and how the ALR has advantages, compared with the CESN and SCR. In Table 1, we summarize the parameter configurations of the CESN, SCR, and ALR for three scenarios, pertaining to reservoir structure and the training/testing strategy. The experiments were undertaken by averaging 50 independent simulation trials.

4.1 Experimental results

4.1.1 NARMA system

The non-linear autoregressive moving average (NARMA) (Jaeger, 2002c; Steil, 2005) system is a discrete time system. Its current output depends on input/output history. Generally speaking, modeling the NARMA system is quite difficult, due to the discretionary non-linearity and possible long memory. The 10th-order NARMA system (Atiya and Parlos, 2000) is given by

Table 1 Experimental setup of the evaluated schemes for the three scenarios

Parameter	Value/Description		
	NARMA system	Chaotic system	Real-world tasks
Reservoir size	[50, 100, 150, 200]	[50, 100, 150, 200]	100
Spectral radius	0.8	0.8	0.8
Output feedback	No	No	No
Output self-feedback	No	No	No
Reservoir activation	tanh	tanh	tanh
Output activation	Linear	Linear	Linear
Data sets (training)	1:2500	1:2500, 1:2500	1:2500, 1:1587
Data sets (testing)	2501:5000	2501:5000, 2501:5000	2501:5000, 1588:3174
Washout time	100	100	500

$$\begin{aligned} \boldsymbol{y}(t) = 0.3\boldsymbol{y}(t-1) + 0.05\boldsymbol{y}(t-1)\sum_{i=1}^{10}\boldsymbol{y}(n-i) \\ +1.5\boldsymbol{x}(t-10)\boldsymbol{x}(t-1)+0.1. \end{aligned} \tag{8}$$

In this task, we use a modified version of the commonly known 10th-order NARMA system, that is,

$$\begin{aligned} \boldsymbol{y}(t) = 0.3\boldsymbol{y}(t-\tau) + 0.05\boldsymbol{y}(t-\tau)\sum_{i=1}^{10}\boldsymbol{y}(n-i\tau) \\ +1.5\boldsymbol{x}(t-\Theta\tau)\boldsymbol{x}(t-\tau)+0.1, \end{aligned} \tag{9}$$

where $\boldsymbol{y}(t)$ is the system output at time t, $\boldsymbol{x}(t)$ is the system input at time t, Θ (Θ=5, 10 in this case) is the order of the NARMA system, and the lag factor τ is used to alter the system delay to control the long-term dependence while keeping the essential structure. For the NARMA tasks with sequence length L=5000, we evaluate the performance of the evaluated schemes with one-step ahead prediction.

Fig. 3 shows the test NRMSEs of the evaluated schemes for the 5th- and 10th-order NARMA datasets, where the proposed ALR obviously outperforms the CESN and SCR, and the SCR exhibits prediction performance comparable to the CESN. Beyond that, for a smaller order NARMA system, the ALR performs much better. The lag factor τ greatly affects the prediction performance of the evaluated schemes, as discussed by Holzmann and Hauser (2010). In all cases, the ALR consistently achieves the best prediction performance with the increase of τ (Table 2). This task demonstrates the possibility of the ALR with simple reservoir structure and only one tuned free parameter in vastly extending the short-term MC, conforming to Lemma 1 which states that the ALR can possess superior MC.

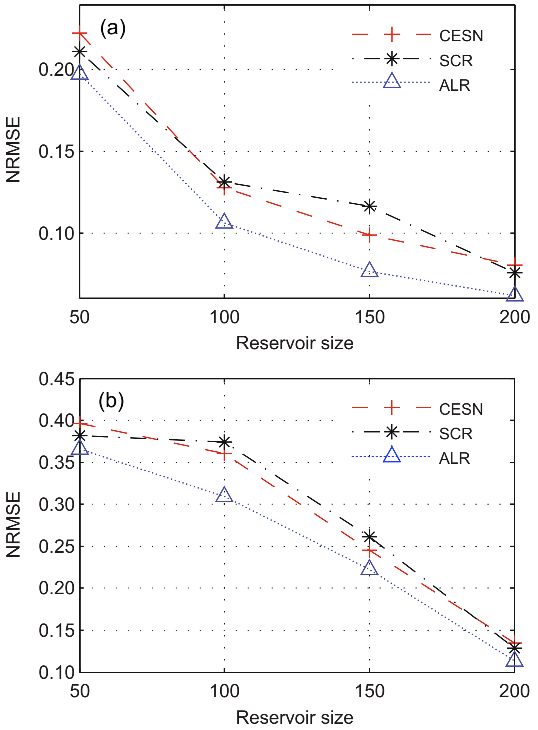

Fig. 3 One-step ahead prediction performance (NRMSE) of the evaluated schemes over a range of the reservoir size
(a) 5th-order NARMA datasets; (b) 10th-order NARMA datasets

Table 2 Prediction performance of the evaluated schemes for different lag factor τ on the 5th-order NARMA dataset[*]

Scheme	NRMSE				
	τ=1	τ=2	τ=3	τ=4	τ=5
CESN	0.0802	0.2019	0.5462	0.7918	0.7977
SCR	0.0753	0.1631	0.4516	0.5281	0.7023
ALR	0.0680	0.1468	0.4223	0.4926	0.6824

[*] Reservoir size N=100

4.1.2 Chaotic system

The Ikeda map is a discrete-time dynamical system given by the nonlinear optical system. It was first proposed by Ikeda *et al.* (1980) as a model of laser light emission from a ring cavity containing a bistable dielectric medium. The Ikeda map dataset is generated by the following form of map:

$$\begin{cases} \boldsymbol{x}(t+1)=1+u\left(\boldsymbol{x}(t)\sin t_n-\boldsymbol{y}(t)\cos t_n\right), \\ \boldsymbol{y}(t+1)=u\left(\boldsymbol{x}(t)\sin t_n+\boldsymbol{y}(t)\cos t_n\right), \end{cases} \tag{10}$$

where $\boldsymbol{y}(t)$ is the Ikeda system output at time t, $\boldsymbol{x}(t)$ is the Ikeda system input at time t. For some values of parameter u (e.g., u=0.7), this system has a chaotic attractor. t_n is a time variable related with $\boldsymbol{y}(t)$ and $\boldsymbol{x}(t)$, given by

$$t_n=0.4-\frac{6}{1+x_n^2+y_n^2}, \tag{11}$$

where the initialization values of system input and output are $\boldsymbol{x}(0)$=0.1 and $\boldsymbol{y}(0)$=0.1, respectively. Meanwhile, the Ikeda map system contains Gaussian white noise with standard deviation v over (0, 1).

The Henon map (Henon, 1976), which maps the (x, y) plane onto itself and exhibits chaotic behavior, is a 2D mapping with a strange attractor. It was created by Hénon as a simplification of the Poincaré section of the Lorenz system, and has been studied extensively due to its ease of computation and interesting behavior. The Henon map is described by

$$\begin{cases} \boldsymbol{x}(t+1)=\boldsymbol{y}(t)+1-a\boldsymbol{x}^2(t), \\ \boldsymbol{y}(t+1)=b\boldsymbol{x}(t), \end{cases} \tag{12}$$

where the Henon map depends on the parameters a and b. To obtain its chaotic behavior, we set a=1.4, b=0.3. Likewise, the Henon map system contains Gaussian white noise with standard deviation v over (0, 1).

In chaotic tasks, we consider one-step ahead predictions of the Ikeda map with L_{Ikeda}=5000 and Henon map with L_{Henon}=5000. The prediction performances of the evaluated schemes are computed for the Ikeda map and Henon map (Fig. 4). The proposed ALR performs better than the considered alternatives, but it does not obtain remarkable improvement in prediction performance. Although our scheme does not dramatically enhance the approximation ability of the chaotic system, the prediction results are encouraging, using such simple ESNs with loop reservoir structure and only one free parameter. Table 3 shows that in noisy chaotic tasks the ALR slightly outperforms the CESN and SCR, and that for relatively small standard deviation *v* the ALR performs much better in the applications, which is attributed to the considerable noise and outliers in the training datasets used.

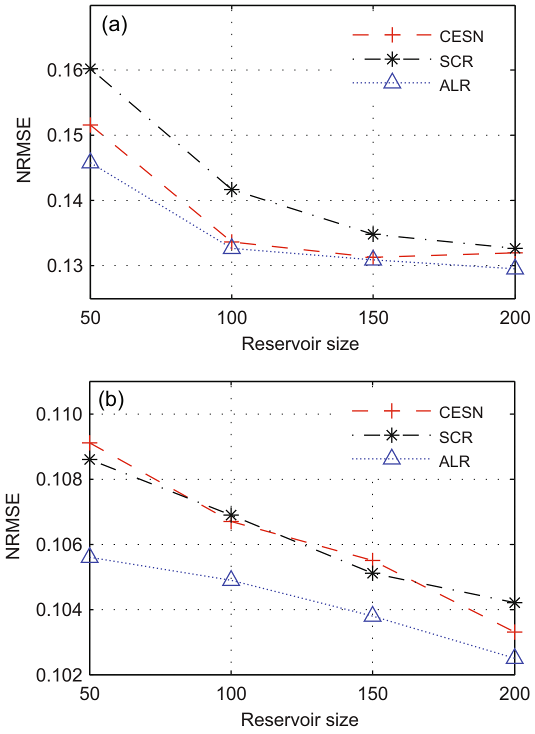

Fig. 4 One-step ahead prediction performance of the evaluated schemes, over a range of the reservoir sizes (noise standard deviation *v*=0.1)
(a) Ikeda map; (b) Henon map

Table 3 Prediction performance of the evaluated schemes for different noise standard deviation *v* on Ikeda map and Henon map*

v	NRMSE					
	Ikeda map			Henon map		
	CESN	SCR	ALR	CESN	SCR	ALR
0.1	0.1337	0.1417	0.1327	0.1067	0.1065	0.1049
0.2	0.2479	0.2532	0.2464	0.2098	0.2102	0.2079
0.3	0.3470	0.3499	0.3459	0.3196	0.3182	0.3151
0.4	0.4475	0.4518	0.4474	0.4337	0.4357	0.4277
0.5	0.5374	0.5419	0.5377	0.5499	0.5491	0.5410
0.6	0.6292	0.6329	0.6296	0.6497	0.6511	0.6346
0.7	0.6944	0.6958	0.6919	0.6982	0.6990	0.6853

* Reservoir size *N*=100

4.1.3 Real-world prediction tasks

Sunspot time series (Schwenker and Labib, 2009): In the experiment, we used the smoothed monthly Sunspot data obtained from National Geophysical Data Center (2007). The Sunspot time series consists of L_{Sunspot}=3154 average smoothed sunspots from January 1749 to October 2011. Based on the input history, we evaluated the one- and four-step ahead prediction performances of the evaluated schemes.

IPIX radar: The sea clutter data obtained by University IPIX radar (the IPIX Radar Sea Clutter Database in McMaster University, available from http://soma.mcmaster.ca/ipix), i.e., radar backscatter from an ocean surface, was regarded as our target input, measured with the McMaster IPIX radar. For the prediction task, L_{Radar}=5000 data was collected to execute the training and testing tasks. Likewise, with one- and four-step ahead predictions, we show whether the proposed scheme has better prediction performance.

Fig. 5 shows the results of the desired and predicted outputs and the error amplitude for the ALR in the real-world Sunspot and IPIX radar tasks. The proposed ALR effectively matches the desired output with prediction output in the IPIX radar task, but in the Sunspot task the result is relatively poor, due to the significant amount of outliers in the Sunspot time series. Table 4 provides the multi-step ahead prediction results of the evaluated schemes. For the Sunspot and IPIX radar tasks, the ALR consistently offers better prediction accuracy over a wide range of choices of the prediction step, compared with the CESN and SCR. Additionally, the prediction performance of the evaluated schemes is highly dependent on the prediction step in the two real-world tasks. For example,

Table 4 Multi-step ahead prediction performance of the evaluated schemes for Sunspot and IPIX radar datasets*

Dataset	Step	NRMSE		
		CESN	SCR	ALR
Sunspot	1	0.3863	0.3876	0.3521
	4	0.5334	0.5247	0.4620
IPIX radar	1	0.2168	0.2143	0.1947
	4	0.4247	0.4305	0.3716

* Reservoir size *N*=100

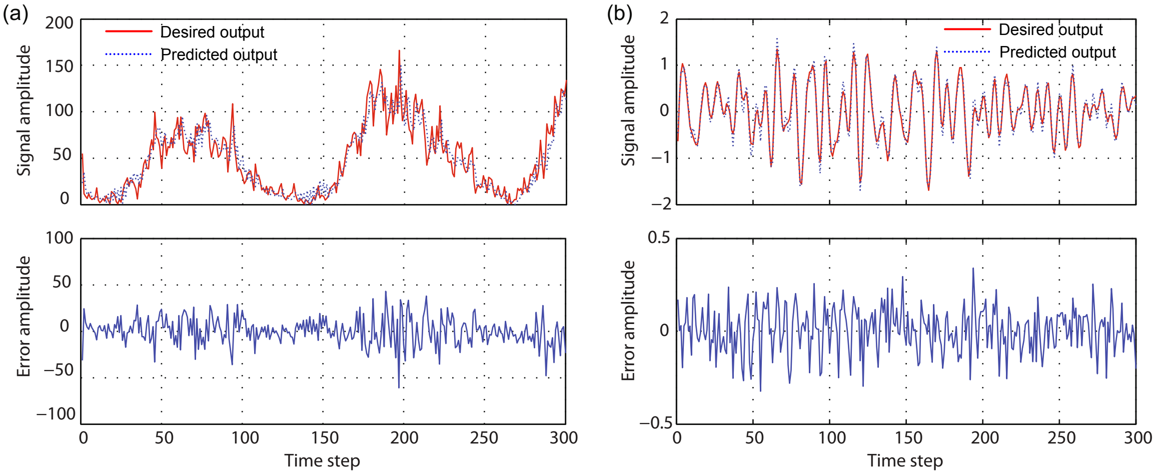

Fig. 5 Output and error of the ALR using one-step ahead prediction (reservoir size *N*=100): (a) Sunspot dataset; (b) Sea Clutter dataset

the ALR scheme using four-step ahead prediction gives rise to about 10% performance degradation compared with that using one-step ahead prediction.

4.2 Performance evaluations

4.2.1 Reservoir size

As shown in Figs. 3 and 4, the reservoir size has considerable effect on the prediction performance of the ALR. Generally speaking, higher prediction accuracy is obtained for a larger reservoir. But the reservoir size cannot be increased infinitely, which leads to an overfitting problem. Hence, for a given task, we gradually increase the reservoir size to avoid prediction performance degradation.

4.2.2 Weight value *r* and spectral radius

In our experiments, the only free parameter *r* is tuned over (0, 1) arbitrarily to satisfy performance requirements of the ALR. Taking the 10th-order NARMA dataset as an example, we analyze how the only free parameter *r*, combined with the spectral radius, significantly impacts the prediction accuracy. As shown in Fig. 6, the prediction accuracy of the ALR increases first and then decreases with the increment of the parameter *r* and spectral radius. The ALR obtains better prediction performance with *r* over [0.5, 0.7] (the optimal value is *r*=0.5, as marked in Fig. 6). This provides the practical guideline for the proper selection of the parameter *r*. Note that although the only free parameter *r* narrows the range of parameter variation compared with two (Rodan and Tino, 2011), our scheme still obtains satisfactory prediction performance, and more importantly, simplifies the trial-and-error process of free parameter selection to enhance flexibility. In other words, this way of parameter configuration can effectively balance the complexity and prediction accuracy. Hence, in different practical tasks, we just need to select the single parameter *r* to guarantee good prediction performance.

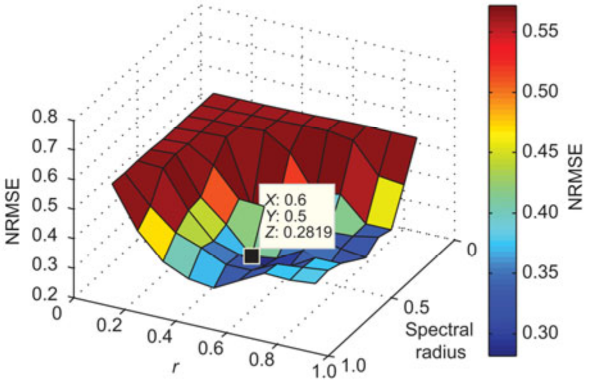

Fig. 6 Weight value *r* versus spectral radius for the ALR on the 10th-order NARMA dataset

The optimal value is *r*=0.5

4.2.3 Input sign pattern

Input sign pattern is also an important parameter that influences the prediction performance. Table 5 shows that among the different input sign generation methods, the prediction performances of the ALR and SCR vary over a range of reservoir sizes for the 10th-order NARMA task. Using the random method and logistic map (Rodan and Tino, 2011), the ALR performs much better than the SCR. Moreover, using the logistic map results in higher prediction accuracy. This shows that the ALR and SCR are sensitive to the input sign pattern in the NARMA task. For chaotic tasks (e.g., Ikeda map and Henon map), the ALR and SCR work similarly well with the input sign generation methods considered, whereas they are relatively insensitive to input sign pattern. Overall, the input sign pattern needs to be determined reasonably for different tasks.

Table 5 Prediction performance of SCR and ALR using different ways of generating input sign

Dataset	Reservoir size	NRMSE			
		SCR, Rand	SCR, Log	ALR, Rand	ALR, Log
10th-order NARMA	50	0.3820	0.3702	0.3658	0.3512
	100	0.3740	0.3487	0.3090	0.2957
	150	0.2613	0.2598	0.2225	0.2091
	200	0.1286	0.1274	0.1136	0.1105
Ikeda map	50	0.0640	0.0638	0.0605	0.0611
	100	0.0335	0.0331	0.0310	0.0315
	150	0.0258	0.0260	0.0195	0.0197
	200	0.0195	0.0200	0.0136	0.0131
Henon map	50	0.1086	0.1093	0.1056	0.1058
	100	0.1069	0.1072	0.1049	0.1054
	150	0.1051	0.1048	0.1038	0.1036
	200	0.1042	0.1045	0.1025	0.1021

Rand: random method; Log: logistic map

4.2.4 Memory capacity

To verify the theoretical result in Lemma 1, we evaluated the MCs of the CESN, SCR, and ALR. Consider that these schemes are trained to recover the k-delay inputs (k=1, 2, ···, 40). For 10th-order NARMA, the following parameters are set: one input unit, 20 reservoir units, 40 output units (one for each k), spectral radius λ=0.8, and fixed weight value r=0.5. What is more, we consider a linear reservoir, in which the units can conduct the identity activation function. Figs. 7a–7c show the k-delay memory capacity of the evaluated schemes. CESN, SCR, and ALR exhibit a close-to-100% recall (detCoeff≈1) for delays up to 15, 12, and 18 respectively, followed by each rugged slope that is still beyond zero at a delay of 40. The detCoeff is the squared correlation coefficient (i.e., MC$_k$ in Eq. (5)), denoting the similarity between the actual and recovered values. The MC values of the corresponding schemes, i.e., the sums of MC$_k$ (k=1, 2, ···, 40), are 18.358, 19.279, and 19.283 for CESN, SCR, and ALR, respectively. Note that the MCs of the SCR and ALR are more or less the same and close to the optimal value 20, but better than that of the CESN. This validates the theoretical result in Lemma 1.

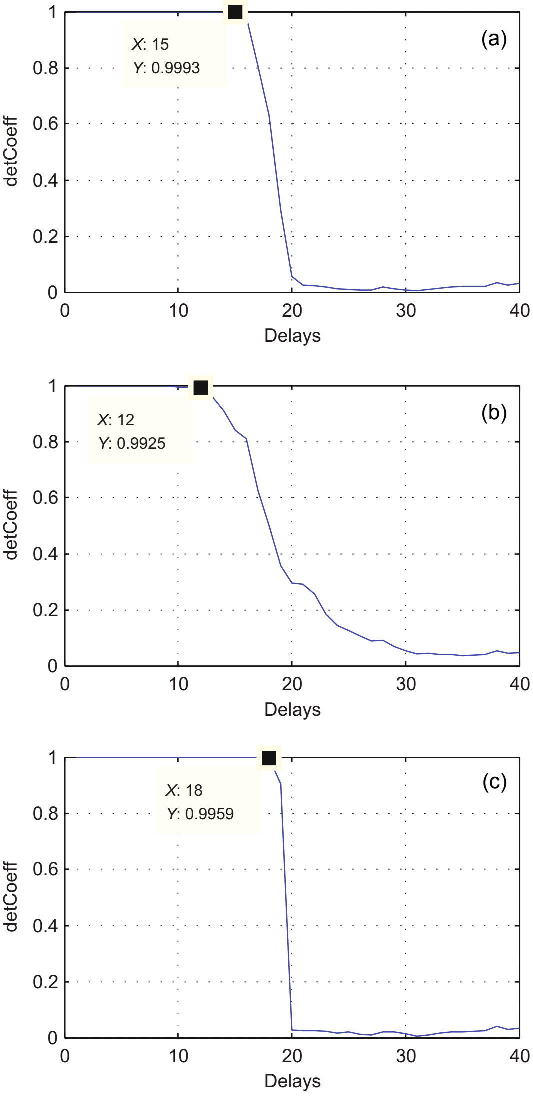

Fig. 7 The forgetting curves of the evaluated schemes in the 10th-order NARMA: (a) CESN; (b) SCR; (c) ALR

4.2.5 Computational complexity

Fig. 8 provides better insight into how well the proposed ALR performs. We compare the training time of the CESN, SCR, and ALR with regard to the computational complexity for the 10th-order NARMA, Henon map, and Sunspot dataset. The CESN requires the most training time, leading to the maximum computational burden, whereas the ALR has the same training time as the SCR, just about half of that in the CESN. For example, the training times of the CESN, SCR, and ALR are 60.8 s, 29.2 s, and 29.3 s for the Henon map task, respectively. This means that the computational burden of the ALR is reduced significantly via the applications of simple loop reservoir structure and the minimization parameter, and is comparable to that of the SCR. Hence, our scheme achieves an effective tradeoff between computational complexity and prediction performance.

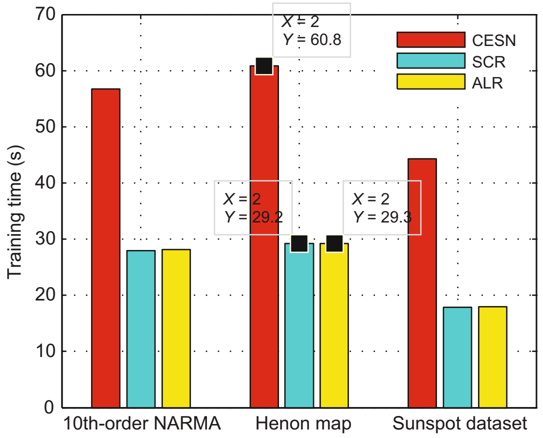

Fig. 8 Comparison of computational complexity of the CESN, SCR, and ALR in the 10th-order NARMA task

5 Conclusions

In this paper, we present a deterministic ESN with loop reservoir structure, namely ALR. The scheme overcomes the limitation of performance degradation for simple ESNs and obtains satisfactory prediction performance. Through theoretical analysis and experimental simulation on both benchmark datasets and real-world tasks, we have shown the following:

1. A deterministic ESN with a loop reservoir is able to achieve better prediction performance than the CESN and SCR.

2. Our scheme requires only one free parameter r be tuned, which considerably simplifies the ESNs. Fewer free parameters can simplify the trial-and-error process for parameter selection and improve system flexibility.

3. We prove that using the same reservoir weight, all the deterministic linear ESNs with the simplest loop reservoir structure (including ALR) have the same optimal MC as the SCR.

In future research, the open issues of the ALR that we will address include exploring multiple-reservoir structure (Xue *et al.*, 2007) to further reduce complexity and examining the utility of different methods for estimating the output weight matrix (Chatzis and Demiris, 2011), since the prediction performance is determined largely by reservoir structure and the estimation method. We will also focus on the MC characteristics for different kinds of input streams.

References

Abbasi Nozari, H., Dehghan Banadaki, H., Mokhtare, M., Hekmati Vahed, S., 2012. Intelligent non-linear modelling of an industrial winding process using recurrent local linear neuro-fuzzy networks. *J. Zhejiang Univ.-Sci. C (Comput. & Electron.)*, **13**(6):403-412. [doi:10.1631/jzus.C11a0278]

Atiya, A.F., Parlos, A.G., 2000. New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE Trans. Neur. Networks*, **11**(3):697-709. [doi:10.1109/72.846741]

Chatzis, S.P., Demiris, Y., 2011. Echo state Gaussian process. *IEEE Trans. Neur. Networks*, **22**(9):1435-1445. [doi:10.1109/TNN.2011.2162109]

Deng, Z.D., Zhang, Y., 2007. Collective behavior of a small-world recurrent neural system with scale-free distribution. *IEEE Trans. Neur. Networks*, **18**(5):1364-1375. [doi:10.1109/TNN.2007.894082]

Henon, M., 1976. A 2-D mapping with a strange attractor. *Commun. Math. Phys.*, **50**(1):69-77. [doi:10.1007/BF01608556]

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., 2001. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies. *In*: Kolen, J., Kremer, S. (Eds.), A Field Guide to Dynamical Recurrent Networks. Wiley-IEEE Press, New York, p.237-243. [doi:10.1109/9780470544037.ch14]

Holzmann, G., Hauser, H., 2010. Echo state networks with filter neurons and a delay&sum readout. *Neur. Networks*, **23**(2):244-256. [doi:10.1016/j.neunet.2009.07.004]

Ikeda, K., Daido, H., Akimoto, O., 1980. Optical turbulence: chaotic behavior of transmitted light from a ring cavity. *Phys. Rev. Lett.*, **45**(9):709-712. [doi:10.1103/PhysRevLett.45.709]

Jaeger, H., 2001. The 'Echo State' Approach to Analysing and Training Recurrent Neural Networks. Technical Report No. 148, German National Research Center for Information Technology, Bremen, Germany.

Jaeger, H., 2002a. A Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the 'Echo State Network' Approach. Technical Report No. GMD 159, German National Research Center for Information

Technology, Sankt Augustin, Germany.

Jaeger, H., 2002b. Short Term Memory in Echo State Networks. Technical Report No. GMD 152, German National Research Center for Information Technology, Sankt Augustin, Germany.

Jaeger, H., 2002c. Adaptive Nonlinear System Identification with Echo State Network. Advances in Neural Information Processing Systems, **15**:593-600.

Jaeger, H., Hass, H., 2004. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science*, **304**(5667):78-80. [doi:10.1126/science.1091277]

Lukosevicius, M., Jaeger, H., 2009. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.*, **3**(3):127-149. [doi:10.1016/j.cosrev.2009.03.005]

Mandic, D.P., Chambers, J.A., 2001. Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability. Wiley, New York, p.31-47. [doi:10.1002/047084535X.ch3]

National Geophysical Data Center, 2007. Sunspot Numbers. Available from http://www.ngdc.noaa.gov/stp/iono/sunspot.html [Accessed on Sept. 23, 2011].

Ozturk, M.C., Principe, J.C., 2007. An associative memory readout for ESNs with applications to dynamical pattern recognition. *Neur. Networks*, **20**(3):377-390. [doi:10.1016/j.neunet.2007.04.012]

Rodan, A., Tino, P., 2011. Minimum complexity echo state network. *IEEE Trans. Neur. Networks*, **22**(1):131-144. [doi:10.1109/TNN.2010.2089641]

Salmen, M., Ploger, P.G., 2005. Echo State Networks Used for Motor Control. Proc. IEEE Int. Conf. on Robotics and Automation, p.1953-1958. [doi:10.1109/ROBOT.2005.1570399]

Schwenker, F., Labib, A., 2009. Echo State Networks and Neural Network Ensembles to Predict Sunspots Activity. 17th European Symp. on Artificial Neural Networks, p.379-384.

Shi, Z.W., Han, M., 2007. Support vector echo-state machine for chaotic time-series prediction. *IEEE Trans. Neur. Networks*, **18**(2):359-372. [doi:10.1109/TNN.2006.885 113]

Siegelmann, H.T., Sontag, E.D., 1991. Turing computability with neural nets. *Appl. Math. Lett.*, **4**(6):77-80. [doi:10.1016/0893-9659(91)90080-F]

Steil, J.J., 2005. Memory in backpropagation-decorrelation *O*(*N*) efficient online recurrent learning. *LNCS*, **3697**: 649-654. [doi:10.1007/11550907_103]

Tino, P., Schittenkopf, C., Dorffner, G., 2001. Financial volatility trading using recurrent neural networks. *IEEE Trans. Neur. Networks*, **12**(4):865-874. [doi:10.1109/72.935096]

Wyffels, F., Schrauwen, B., Stroobandt, D., 2008. Stable output feedback in reservoir computing using ridge regression. *LNCS*, 5163:808-817. [doi:10.1007/978-3-540-87536-9_83]

Xia, Y.L., Jelfs, B., van Hulle, M.M., Principe, J.C., Mandic, D.P., 2011. An augmented echo state network for nonlinear adaptive filtering of complex noncircular signals. *IEEE Trans. Neur. Networks*, **22**(1):74-83. [doi:10.1109/TNN.2010.2085444]

Xue, Y.B., Yang, L., Haykin, S., 2007. Decoupled echo state networks with lateral inhibition. *Neur. Networks*, **20**(3): 365-376. [doi:10.1016/j.neunet.2007.04.014]

Zhang, B., Miller, D.J., Wang, Y., 2012. Nonlinear system modeling with random matrices: echo state networks revisited. *IEEE Trans. Neur. Networks Learn. Syst.*, **23**(1): 175-182. [doi:10.1109/TNNLS.2011.2178562]

Appendix: Proof of Lemma 1

For all the linear ESNs with the loop reservoir structure, we first define the input vector $\boldsymbol{W}^{\text{in}}=(W_1, W_2, \cdots, W_N)^{\text{T}}$ and unit feedback vector $\boldsymbol{H}_k=(W_k, S_{k,1}, S_{k,2}, \cdots, S_{k,N-1})^{\text{T}}$. Then, the regular matrix $\boldsymbol{\Psi}$ is given by $\boldsymbol{\Psi}=(\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_N)$. Moreover, we denote a diagonal matrix by $\boldsymbol{\Lambda}$ ($\boldsymbol{\Lambda}=\text{diag}\{1, r, r^2, \cdots, r^{N-1}\}$) and the matrix $\boldsymbol{\Psi}^{\text{T}}\boldsymbol{\Lambda}^2\boldsymbol{\Psi}$ by $\boldsymbol{A}$ ($\boldsymbol{A}=\boldsymbol{\Psi}^{\text{T}}\boldsymbol{\Lambda}^2\boldsymbol{\Psi}$). Let $\boldsymbol{A}$ be an invertible matrix. According to the Lemma 1 in Rodan and Tino (2011), we have

$$\lambda_k = \boldsymbol{H}_k^{\text{T}}\boldsymbol{A}^{-1}\boldsymbol{H}_k = \begin{cases} r^{-2k}, & k=1,2,\ldots,N-1, \\ 1, & k=N. \end{cases} \tag{A1}$$

Consider the following conditions: an i.i.d. zero-mean single-channel input stream $\cdots\boldsymbol{u}(t-1)\boldsymbol{u}(t)$, linear reservoir (i.e., reservoir activation is the identity function), and the same reservoir weight r. For example, when the i.i.d. zero-mean single-channel input stream is injected to the ALR, we show the activations of all reservoir units at time t as follows:

$$\begin{aligned} x_1(t) = {} & W_1u(t) + r(W_N + W_2)u(t-1) \\ & + r^2(W_{N-1} + W_3)u(t-2) + r^3(W_{N-2} + W_4)u(t-3) \\ & + \cdots + r^{N-1}(W_2 + W_N)u(t-(N-1)) \\ & + r^N W_1 u(t-N) + r^{N+1}(W_N + W_2)u(t-(N+1)) + \cdots \\ & + r^{2N-1}(W_2 + W_N)u(t-(2N-1)) + r^{2N}W_1u(t-2N) \\ & + r^{2N+1}(W_N + W_2)u(t-(2N+1)) + \cdots \\ x_2(t) = {} & W_2u(t) + r(W_1 + W_3)u(t-1) \\ & + r^2(W_N + W_4)u(t-2) + r^3(W_{N-1} + W_5)u(t-3) \\ & + \cdots + r^{N-1}(W_3 + W_1)u(t-(N-1)) \\ & + r^N W_2 u(t-N) + r^{N+1}(W_1 + W_3)u(t-(N+1)) + \cdots \\ & + r^{2N-1}(W_3 + W_1)u(t-(2N-1)) + r^{2N}W_2u(t-2N) \\ & + r^{2N+1}(W_1 + W_3)u(t-(2N+1)) + \cdots \\ & \vdots \end{aligned}$$

$$\begin{aligned}x_N(t) &= W_N u(t) + r(W_{N-1}+W_1)u(t-1)\\ &\quad + r^2(W_{N-2}+W_2)u(t-2) + r^3(W_{N-3}+W_3)u(t-3)\\ &\quad + \cdots + r^{N-1}(W_1+W_{N-1})u(t-(N-1))\\ &\quad + r^N W_N u(t-N) + r^{N+1}(W_{N-1}+W_1)u(t-(N+1))\\ &\quad + \cdots + r^{2N-1}(W_1+W_{N-1})u(t-(2N-1))\\ &\quad + r^{2N}W_N u(t-2N)\\ &\quad + r^{2N+1}(W_{N-1}+W_1)u(t-(2N+1)) + \cdots\end{aligned}$$

For the first unit activation $x_1(t)$, W_N+W_2, W_{N-1}+W_3, ···, W_2+W_N denote the sum of the input weights of the units with j delays (j=1, 2, ···, N−1) from the first reservoir unit, respectively. By analogy, the other unit activations have similar expressions. Furthermore, for all the ESNs with the simplest loop reservoir structure, the unit activations are formulated as

$$\begin{aligned}x_k(t) &= W_k u(t) + rS_{k,1}u(t-1)\\ &\quad + r^2 S_{k,2}u(t-2) + r^3 S_{k,3}u(t-3)\\ &\quad + \cdots + r^{N-1}S_{k,N-1}u(t-(N-1))\\ &\quad + r^N W_k u(t-N) + r^{N+1}S_{k,1}u(t-(N+1)) \quad \text{(A2)}\\ &\quad + \cdots + r^{2N-1}S_{k,N-1}u(t-(2N-1))\\ &\quad + r^{2N}W_k u(t-2N)\\ &\quad + r^{2N+1}S_{k,1}u(t-(2N+1)) + \cdots,\end{aligned}$$

and $S_{k,j}$ is given by

$$S_{k,j} = \sum_{i=1}^{N} W_i \quad \forall j = 1,2,\ldots,N-1,\ k = 1,2,\ldots,N, \quad \text{(A3)}$$

where $S_{k,j}$ is described as the sum of the input weights of the units with j delays from the kth reservoir unit, and W_i is defined as follows: if there is a delay of j units between the kth and ith (i=1, 2, …, N) reservoir units, then W_i≠0; otherwise, W_i=0.

To prove the memory capacity, the optimal output weight can be first calculated using the Wiener-Hopf equation:

$$\boldsymbol{W}^{\text{out}} = \boldsymbol{R}^{-1}\boldsymbol{p}_k, \quad \text{(A4)}$$

where $\boldsymbol{R}$=$E[\boldsymbol{x}(t)\boldsymbol{x}(t)^{\mathrm{T}}]$ is the N×N correlation matrix of different reservoir states, and $\boldsymbol{p}_k$=$E[\boldsymbol{x}(t)\boldsymbol{u}(t-k)]$ is the N×1 cross-correlation matrix of reservoir states with desired output.

Then, for the i.i.d. zero-mean single-channel input stream, according to Eq. (1), the reservoir activation is updated. We can obtain the covariance matrix $\boldsymbol{R}$ by analogy. Firstly, the internal element R_{12} can be calculated by

$$\begin{aligned}R_{12} &= E[\boldsymbol{x}(t)\boldsymbol{x}(t)^{\mathrm{T}}]\\ &= E[W_1W_2u^2(t) + r^2S_{1,1}S_{2,1}u^2(t-1)\\ &\quad + r^4S_{1,2}S_{2,2}u^2(t-1) + \cdots\\ &\quad + r^{2(N-1)}S_{1,N-1}S_{2,N-1}u^2(t-(N-1))\\ &\quad + r^{2N}W_1W_2u^2(t-N)\\ &\quad + r^{2(N+1)}S_{1,1}S_{2,1}u^2(t-(N+1)) + \cdots\\ &\quad + r^{2(2N-1)}S_{1,N-1}S_{2,N-1}u^2(t-(2N-1))\\ &\quad + r^{4N}W_1W_2u^2(t-2N) + \cdots]\\ &= W_1W_2\mathrm{Var}[u(t)] + r^2S_{1,1}S_{2,1}\mathrm{Var}[u(t-1)]\\ &\quad + r^4S_{1,2}S_{2,2}\mathrm{Var}[u(t-2)] + \cdots\\ &\quad + r^{2N}W_1W_2\mathrm{Var}\left[u(t-N)\right] + \cdots\\ &= \sigma^2(W_1W_2 + r^2S_{1,1}S_{2,1} + r^4S_{1,2}S_{2,2} + \cdots\\ &\quad + r^{2(N-1)}S_{1,N-1}S_{2,N-1})\sum_{j=0}^{\infty} r^{2Nj},\end{aligned}$$

where σ^2 is the variance of the input stream. The above equation is simplified as

$$R_{1,2} = \frac{\sigma^2}{1-r^{2N}}\boldsymbol{H}_1^{\mathrm{T}}\boldsymbol{\varLambda}^2\boldsymbol{H}_2. \quad \text{(A5)}$$

By analogy,

$$R_{i,j} = \frac{\sigma^2}{1-r^{2N}}\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{\varLambda}^2\boldsymbol{H}_j, \quad i,j = 1,\ 2,\ \ldots,\ N, \quad \text{(A6)}$$

and then

$$\boldsymbol{R} = \frac{\sigma^2}{1-r^{2N}}\boldsymbol{\varPsi}^{\mathrm{T}}\boldsymbol{\varLambda}^2\boldsymbol{\varPsi} = \frac{\sigma^2}{1-r^{2N}}\boldsymbol{A}. \quad \text{(A7)}$$

Likewise, $\boldsymbol{p}_k$ can be evaluated by

$$\boldsymbol{p}_k = \sigma^2 r^k \boldsymbol{H}_k. \quad \text{(A8)}$$

According to Eq. (A4), for all the ESNs with the simplest loop reservoir structure, the general optimal output weight is given by

$$\boldsymbol{W}^{\text{out}}=(1-r^{2N})r^{k}\boldsymbol{A}^{-1}\boldsymbol{H}_{k}. \tag{A9}$$

Furthermore, we can easily obtain the general ESN output at time step t:

$$\boldsymbol{y}(t)=\boldsymbol{x}(t)^{\text{T}}\boldsymbol{W}^{\text{out}}=(1-r^{2N})r^{k}\boldsymbol{x}(t)^{\text{T}}\boldsymbol{A}^{-1}\boldsymbol{H}_{k}. \tag{A10}$$

Then, we can compute the covariance of $\boldsymbol{y}(t)$ with $\boldsymbol{u}(t-k)$ as

$$\begin{aligned}
&\text{Cov}(\boldsymbol{y}(t),\boldsymbol{u}(t-k))\\
&=r^{k}(1-r^{2N})\text{Cov}(\boldsymbol{x}(t)^{\text{T}},\boldsymbol{u}(t-k))\boldsymbol{A}^{-1}\boldsymbol{H}_{k}\\
&=r^{k}(1-r^{2N})[E(\boldsymbol{x}(t)^{\text{T}}\boldsymbol{u}(t-k))\\
&\quad-E(\boldsymbol{x}(t)^{\text{T}})E(\boldsymbol{u}(t-k))]\boldsymbol{A}^{-1}\boldsymbol{H}_{k}\\
&=r^{k}(1-r^{2N})E(\boldsymbol{x}(t)^{\text{T}}\boldsymbol{u}(t-k))\boldsymbol{A}^{-1}\boldsymbol{H}_{k}\\
&=\sigma^{2}r^{k}(1-r^{2N})\boldsymbol{H}_{k}^{\text{T}}\boldsymbol{A}^{-1}\boldsymbol{H}_{k}\\
&=\sigma^{2}r^{k}(1-r^{2N})\lambda_{k},
\end{aligned}$$

and the variance of $\boldsymbol{y}(t)$ as

$$\begin{aligned}
&\text{Var}(\boldsymbol{y}(t))\\
&=E[(\boldsymbol{W}^{\text{out}})^{\text{T}}\boldsymbol{x}(t)\boldsymbol{x}(t)^{\text{T}}\boldsymbol{W}^{\text{out}}]-E^{2}[\boldsymbol{x}(t)^{\text{T}}\boldsymbol{W}^{\text{out}}]\\
&=(\boldsymbol{W}^{\text{out}})^{\text{T}}E[\boldsymbol{x}(t)\boldsymbol{x}(t)^{\text{T}}]\boldsymbol{W}^{\text{out}}\\
&=(\boldsymbol{W}^{\text{out}})^{\text{T}}\boldsymbol{R}\boldsymbol{W}^{\text{out}}\\
&=\boldsymbol{p}_{k}^{\text{T}}\boldsymbol{R}^{-1}\boldsymbol{p}_{k}\\
&=\sigma^{2}r^{2k}(1-r^{2N})\boldsymbol{H}_{k}^{\text{T}}\boldsymbol{A}^{-1}\boldsymbol{H}_{k}\\
&=\text{Cov}(\boldsymbol{y}(t),\boldsymbol{u}(t-k)).
\end{aligned}$$

Using Eq. (5), the k-delay short memory capacity of all the linear ESNs with the simplest loop reservoir is given by

$$\begin{aligned}
\text{MC}_{k}&=\frac{\text{Cov}^{2}(\boldsymbol{y}(t),\boldsymbol{u}(t-k))}{\text{Var}(\boldsymbol{y}(t))\text{Var}(\boldsymbol{u}(t))}\\
&=\frac{\text{Cov}(\boldsymbol{y}(t),\boldsymbol{u}(t-k))}{\text{Var}(\boldsymbol{u}(t))}\\
&=r^{2k}(1-r^{2N})\lambda_{k}.
\end{aligned} \tag{A11}$$

From Eq. (A11), we observe that all deterministic ESNs evolving from the simplest loop reservoir structure have the same k-delay MC as the SCR. Thus, according to Eq. (6), they have the same memory capacity as the SCR, that is MC=N−1+r^{2N}.