

Complex Systems Modeling Using Scale-Free Highly-Clustered Echo State Network

Zhidong Deng, *Member, IEEE*, and Yi Zhang

Abstract—Inspired by the universal laws governing different kinds of complex networks, we propose a scale-free highly-clustered echo state network (SHESN). Different from echo state network (ESN), the state reservoir of the SHESN is generated by natural growth rules and eventually forms a complex network with small-world, scale-free properties, and hierarchically distributed structure. We implemented a large-scale SHESN with 3,000 internal neurons and applied it to modeling the pH-neutralization process. Simulation results showed the superior performance of SHESN. Furthermore, we analyzed the natural characteristics of the SHESN and discussed our growth rules and the new state reservoir from a brain functional network perspective.

I. INTRODUCTION

Recently, complex network theory is fascinatingly advancing. In a large number of real-world natural and artificial complex networks, such as the neural network of *C. Elegans*, cellular and metabolic networks, food web, the World-Wide Web, the BBS, the Internet backbone, power grid, citation network, and many social networks, small-world phenomena and scale-free properties have ubiquitously been discovered in the past few years [1], [2], [6], [11], [18]–[20]. The small-world network first introduced by Watts and Strogatz [20] is defined as a highly clustered network with a short average characteristic path length, and was conceptually originated from Milgram’s work in the late 1960s, or the so-called “six degrees of separation” principle [15]. The scale-free network has a power law form that is independent of the network scale [1]. The small-world effect and scale-free property, which are viewed as universal laws governing different kinds of complex networks, including biological networks [2] and brain functional networks [4], [20], has led dramatic advances in exploring large-scale complex networks.

On the other hand, biological neuronal system, which typically has a massive number of neurons, recurrent pathways, and sparse connectivity [8], [9], are very different from most of the existing artificial neural networks such as BP networks [17], Hopfield networks [7], and Elman networks [5]. Interestingly, H. Jaeger and his co-authors proposed a new recurrent neural network (RNN), called echo state networks (ESNs), for function approximation, chaotic time series prediction, and modeling nonlinear dynamic systems [8], [9]. The ESN model contains a completely random state reservoir as a hidden layer, which is usually composed

of hundreds or thousands of internal neurons. This very promising RNN model reflects some features of biological brain and the resulting learning algorithm is computationally efficient and easy to use.

Inspired by both complex network theory and biological neuronal system, this paper proposes an extended ESN model that contains a naturally evolving state reservoir, which has scale-free property, small-world effect, and hierarchically distributed structure. The new state reservoir is incrementally generated using our natural growth rules and eventually comprises a collection of domains where a large number of local internal neurons are sparsely interconnected. Essentially, such a scale-free highly-clustered echo state network (SHESN) interpolates between a completely regular Elman network and a completely random ESN. In our experiment, we have implemented the SHESN that contains up to 3,000 internal neurons. Experimental results showed that our SHESN is capable of accurately approximating the inverse dynamic model of the pH-neutralization process. Furthermore, the collective behavior of the proposed RNN model was empirically analyzed and discussed. We revealed that the SHESN has many interesting natural characteristics that were extensively present in a variety of artificial and natural complex networks, e.g., power law distribution for network degree, short average characteristic path length, high clustering coefficient, and spatially hierarchical and distributed structure.

The rest of the paper is organized as follows. In Section II, we proposed our SHESN, including the network structure, the incremental growth rules used for producing a naturally evolving state reservoir, and the one-step learning algorithm. In Section III, we applied the SHESN to modeling the inverse dynamic system of the pH-neutralization process. The high nonlinearity and sensitivity of the pH-neutralization problem was analyzed, experimental results were achieved, and comparative study was done. In Section IV, we empirically analyzed a small-world feature and a scale-free characteristic of the naturally evolving state reservoir, and discussed the relationship between our new state reservoir and brain functional network. In Section V, we concluded this paper.

II. SCALE-FREE HIGHLY-CLUSTERED ECHO STATE NETWORK

The architecture of scale-free highly-clustered echo state network (SHESN) is shown in Fig. 1, which is composed of three layers: an input layer, a new state reservoir (or a hidden layer), and an output layer. The state reservoir is generated

Zhidong Deng is with the Department of Computer Science, Tsinghua University, Beijing 100084, China (email: michael@tsinghua.edu.cn).

Yi Zhang is with the Department of Computer Science, Tsinghua University, Beijing 100084, China (email: zhang-yi@mails.tsinghua.edu.cn).

The two authors contribute equally to this paper.

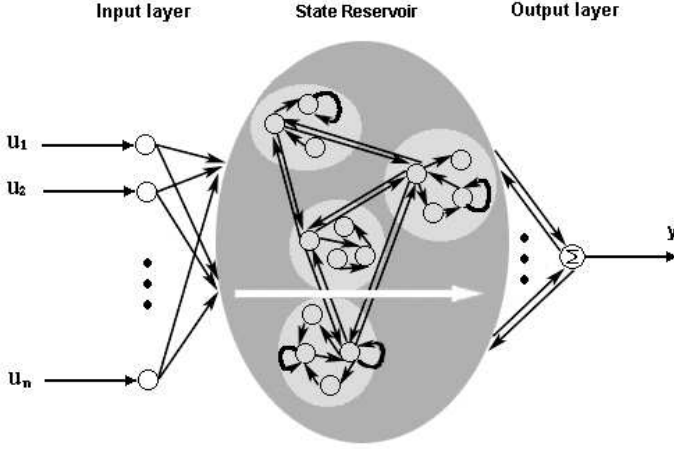


Fig. 1. Network Architecture of the SHESN. It consists of an input layer, a new state reservoir, and an output layer. The generated state reservoir has a collection of naturally evolving features.

by our natural growth rules. All the internal neurons in the resulting state reservoir are geographically scattered on a grid plane and eventually form a hierarchical and distributed structure.

A. Network Architecture of the SHESN

At time step k , the input layer receives an r -dimensional input vector $u(k) = [u_1(k), u_2(k), \dots, u_r(k)]^T$ and passes it to all the internal neurons in the new state reservoir through an $n \times r$ input weight matrix W^{in} . The i th neuron in this layer is directly connected to the i th component $u_i(k)$ of input vector $u(k)$. The output of the i th input neuron simply equals to its corresponding input $u_i(k)$. In our new state reservoir, all n internal neurons are sparsely interconnected using an $n \times n$ reservoir weight matrix W^{res} . The output of each internal neuron is called a state, and all the outputs are denoted by $x(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T$. In order to produce an m -dimensional output vector $y(k) = [y_1(k), y_2(k), \dots, y_m(k)]^T$, the neurons in the output layer collect all the states $x_i(k)$ ($i = 1, 2, \dots, n$) in the preceding new state reservoir along with all the inputs $u_i(k)$ ($i = 1, 2, \dots, r$) in the first layer, through an $m \times (n + r)$ output weight matrix W^{out} , which will be discussed later. Meanwhile, the network output vector $y(k)$ is fed back to all the internal neurons through an $n \times m$ feedback weight matrix W^{fb} . The activation functions in the last two layers are all set to $\tanh(\cdot)$. Just like that of [9], input and feedback weights of W^{in} and W^{fb} are randomly assigned with uniform distribution, and the output weight matrix W^{out} is adjusted using supervised learning. But the reservoir weight W^{res} is produced according to our naturally evolving rules, rather than completely random approaches in [8], [9].

The forward propagation of the SHESN is given below:

$$x(k+1) = \tanh(W^{res}x(k) + W^{in}u(k+1) + W^{fb}y(k) + v(k))$$

$$y(k) = \tanh\left(W^{out} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}\right)$$

where $v(k)$ is a noise (threshold) that is added to the activation functions of internal neurons.

B. Natural Growth Rules: Generating a New State Reservoir of the SHESN

Our natural-growth rules of generating a new state reservoir roughly involve the following six steps:

- a) Initialize an $L \times L$ grid plane of a state reservoir.
- b) Put backbone neurons on the grid plane and then generate synaptic connections among the backbone neurons.
- c) Add one new local neuron to the grid plane.
- d) Produce synaptic connections for the newly added local neuron using local preferential attachment rules.
- e) Repeat steps c) and d) for each new local neuron.
- f) Generate a reservoir weight matrix W^{res} such that the echo state property holds [8].

We will describe the above natural-growth rules in detail below.

1) *Initialization of an $L \times L$ Grid Plane of a State Reservoir:* Let the number of internal neurons in a new state reservoir be n , where $n \ll L \times L$. Using a stochastic dynamic growth model like that proposed by [14], the internal neurons with a size of n are incrementally assigned on a grid plane of state reservoir divided into $L \times L$ squares. The $L \times L$ grid plane or the reservoir with a capacity of $L \times L$ is required to be large enough in order to contain all the n internal neurons ($n \ll L \times L$). Note that different internal neurons gradually generated can not be placed at the same location on the grid plane of the reservoir and are not allowed beyond the grid plane.

2) *Generation of Backbone Neurons and Synaptic Connections Associated:* In the process of generating a new state reservoir, we classified all the internal neurons into two categories: backbone neurons and local neurons. Usually, the number of backbone neurons must be much less than that of local neurons. In our experiment to be described later, for example, about one percent of internal neurons in the new state reservoir were backbone ones. Specifically, we randomly generated the x - and y -coordinates $[x_b, y_b]$ of each backbone neuron on the grid plane. A fixed set of backbone neurons was then assigned at different geographical locations on the $L \times L$ grid plane, which roughly determined the geographical distribution of all n internal neurons. Furthermore, we defined a domain as the set of internal neurons that comprises one backbone neuron and a number of local neurons around this backbone. A collection of domains constituted our new state reservoir from a higher level perspective.

The geographical distribution of n_b backbone neurons, however, must satisfy two restrictions. One is that different backbone neurons generated are not allowed to be at the same location on the grid plane of the reservoir. The other is that the minimum distance between any two backbone neurons must be greater than a certain threshold such that the resulting domains could be separated from each other. For instance, we set this threshold to be 30 in our experiment. After that, the backbone neurons of n_b were fully connected

to each other through synaptic connection weights that were randomly set to be a real value between $[-1, 1]$.

3) *Incremental Growth of New Local Neurons:* As described above, we initially built a small fully connected backbone network, which consisted of n_b backbone neurons. Other (local) neurons were then generated so as to form an entire naturally evolving state reservoir. In this process, for each local neuron, we randomly selected one of the backbone neurons and put the local neuron into the domain associated with this backbone. Suppose that the location of this backbone neuron is $[x_b, y_b]$. We then produce the x- and y-coordinates of local neurons around the backbone one $[x_b, y_b]$ on the grid plane according to the bounded Pareto heavy-tailed distribution [3]:

$$P(v) = \frac{ak^a}{1 - (k/Q)^a} v^{-a-1}, \quad k \leq v \leq Q.$$

where a denotes the shape parameter and $k(Q)$ the minimum (maximum) value.

If a local neuron is farther from its own backbone neuron than one of the other backbone neurons, we assume this local neuron is located in another domain dominated by the nearest backbone neuron, which enables the incremental growth of new local neurons to avoid collisions. But this case, fortunately, happens rarely in the procedure presented here due to the fact that the bounded Pareto distribution seldom generates a high value [3].

As a result, it causes a popular and natural effect that in the same domain, or equivalent cortex in a sense, most of the local neurons are geographically near to their backbone, while a few of them are far away from it. Such a geographical distribution of internal neurons bears a resemblance to that present in the human brain network [4].

4) *Generation of Synaptic Connections for New Local Neuron Using Local Preferential Attachment Rules:* Based on preferential attachment rules [1], any newly added local neurons always prefer to connect to neurons that already have many synaptic connections. More precisely, the probability that a new local neuron is connected to an existing neuron is proportional to the outdegree of the existing neuron. Considering the domain concept introduced here, we present a new strategy, henceforth called a local preferential attachment.

For convenience, we refer to the domain that contains a new local neuron as the current domain. The candidate neighborhood of a new local neuron is defined as the set of neurons to which this new local neuron is allowed to be connected. Specifically, assume that there exists a circle whose center is the location of the new local neuron and the radius is the Euclidean distance from the new local neuron to the backbone neuron in the current domain. Consequently, we choose all those neurons in the current domain that are just within such a circle as the candidate neighborhood of the new local neuron. Apparently, the backbone neuron is always one of the candidate neighbors farthest from the new local neuron in the current domain.

Let n_c represent the number of synaptic connections for a newly added local neuron. The parameter n_c controls

the density of connections generated in the current domain. Additionally, let n_1 and n_2 be the number of neurons, respectively, in the current domain and in the candidate neighborhood of a new local neuron ($n_1 \geq n_2$).

Our local preferential attachment rules are given as follows:

- (a) If $n_c \geq n_1$, a new local neuron is fully connected to all the existing neurons in the current domain.
- (b) If $n_2 \leq n_c < n_1$, the candidate neighborhood is redefined as the set of all the neurons in the current domain, instead of just those neurons in the circle specified above. A new local neuron is connected to all the candidate neighbors using the following probability [14]:

$$\frac{d_i}{\sum_{j \in C} d_j},$$

where d_j is the current outdegree of neuron j , and C the candidate neighborhood of the new local neuron.

- (c) If $n_c < n_2$, the probability that is used to attach a new local neuron to a candidate neighbor is the same as the above.

In general, local preferential attachment rules contribute to the scale-free property of generated networks. The chosen candidate neighbors of a new local neuron, of which the new local neuron itself is usually in the center, should help improve the clustering coefficient of the network.

5) *Spatially Hierarchical and Distributed Structure:* As shown in Fig. 2, the 3,000 internal neurons are incrementally placed on the 500×500 grid plane, resulting in 30 clear clusters or domains. Each domain contains one backbone neuron, and a collection of local neurons around the backbone one. It is readily evident that the network topology of our naturally evolving reservoir has a geographically hierarchical and distributed structure at different levels. It has several unique features as follows:

- (a) The new dynamic state reservoir comprises several domains. The domains are fully connected to each other through backbone neurons and can be regarded as top level macro-neurons in the reservoir network hierarchy.
- (b) In each domain or at a low level, local neurons are only connected to the neurons located in the same domain. In most cases, the number of intra-domain connections is much greater than that of inter-domain ones.
- (c) According to (a) and (b), dynamic behaviors for each domain are relatively independent.

C. Supervised Learning

As mentioned above, the reservoir weight matrix, i.e., W^{res} , must be carefully chosen in order to ensure that echo state property holds [8]. Meanwhile, the input weight matrix W^{in} and the feedback weight matrix W^{fb} could be arbitrarily assigned within the range of possibilities. But the output weight matrix W^{out} must be adjusted using supervised learning.

For instance, the training dataset of 1,001 length is defined as follows:

$$\{u(1); y_d(1)\}, \{u(2); y_d(2)\}, \dots, \{u(1001); y_d(1001)\}$$

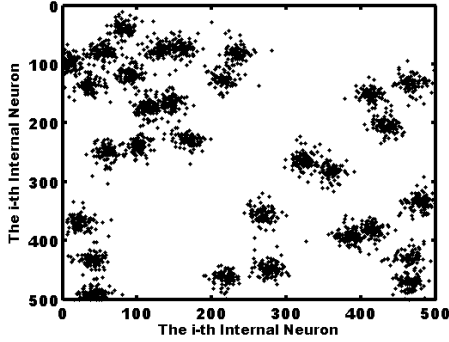


Fig. 2. The 3,000 internal neurons in a naturally evolving state reservoir are geographically distributed on an 500×500 grid plane.

where $u(k)$ denotes the input vector of SHESN at time k and $y_d(k)$ the desired output.

After discarding the first 50 steps or going through the transient time [9], we must find the output weight matrix W^{out} such that the training mean-square error (*MSE*):

$$MSE = \frac{1}{951} \sum_{k=51}^{1001} \left(d(k) - W^{out} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \right)^2$$

is minimized. Note that $d(k) = (\tanh)^{-1}(y_d(k))$, $x(k) = [x_1(k) \ x_2(k) \ \dots \ x_n(k)]^T$, and n indicates the number of echo state variables.

Apparently, this is a classical linear regression problem. It is easy to directly solve it using a generalized inverse matrix approach. Therefore the $1 \times (n+1)$ dimensional output matrix W^{out} is derived as follows:

$$W^{out} = (M^{-1}D)^T$$

where T represents the transpose. The $951 \times (n+1)$ dimensional matrix M is given by

$$M = \begin{bmatrix} x_1(51) & \dots & x_n(51) & u(51) \\ x_1(52) & \dots & x_n(52) & u(52) \\ \dots & \dots & \dots & \dots \\ x_1(1001) & \dots & x_n(1001) & u(1001) \end{bmatrix}$$

and

$$D = [d(51) \ d(52) \ \dots \ d(1001)]^T$$

In our experiment, calculations of a generalized inverse matrix of M were done by using the *pinv* pseudoinverse function in MATLAB.

III. APPLICATIONS TO THE PH-NEUTRALIZATION PROCESS

The modeling and control of the pH-neutralization process is a challenging problem [13]. In order to validate performance of the proposed network model, we apply the SHESN to building the inverse model of this highly nonlinear and quite sensitive dynamic system. The structure scheme of the inverse dynamic modeling used in this paper is shown in Fig. 3, where the output and the input of the pH-neutralization process are used as the input and the desired output of the SHESN, respectively.

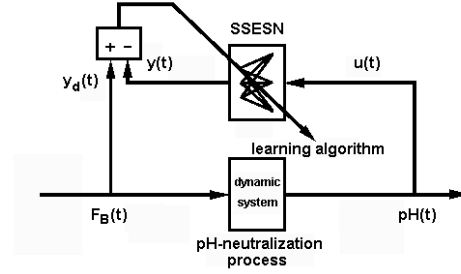


Fig. 3. The modeling structure scheme of the pH-neutralization processes.

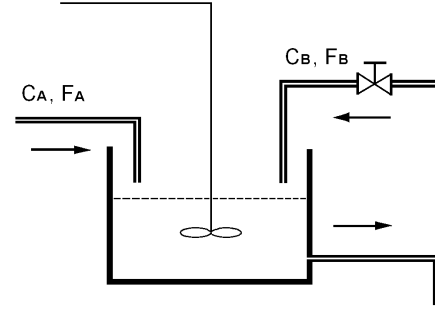


Fig. 4. Schematic diagram of the pH-neutralization process.

A. Problem Description

As shown in Fig. 4, the pH-neutralization process neutralizes strong acid with strong base in a continuous stirred tank reactor. The control objective is to keep the *pH* value at the point of neutrality ($pH = 7$) in the outlet stream from the tank using the control of the flow of base. This process has widely been found in many fields such as chemical industry and waste water treatment.

In fact, the pH-neutralization process is quite difficult to fulfill an exact modeling and control. First, the process is highly nonlinear. Second, it is very sensitive to disturbance near the point of neutrality. Finally, it is difficult to formulate and to identify a mathematical model for the pH-neutralization process because even a small amount of polluting elements, e.g., carbonate or phosphate with buffering capability, leads to a considerable change in dynamics of the process.

Under an ideal condition, i.e., no polluting elements occurred in the reactor, the nonlinear state equation of the pH-neutralization process is described by a quasi-linear equation as follows [12]:

$$V \frac{dQ}{dt} = - (F_A + F_B) Q + C_A F_A - C_B F_B,$$

where $Q = [H^+] - [OH^-]$ is the difference between two ion concentrations, which indicates a deviation from neutrality. Apparently, Q equals to zero at the neutrality point $pH = 7$. $C_A(F_A)$ and $C_B(F_B)$ denote a concentration (a flow) of acid and base, respectively, and V a tank volume. In our experiment, we selected parameters as: $C_A = C_B = 10^{-3}$ (mol/l), $F_A = 0.110$ (l/min), and $V = 2.0$ (l).

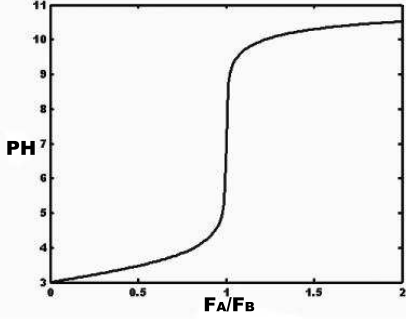


Fig. 5. Neutralization curve as $C_A = C_B = 10^{-3}$ (mol/l)

In industrial practice, unfortunately, direct measurements are normally on the pH value rather than Q . For this reason, Q can be calculated from the measured pH value. But the procedure might require the great accuracy of computation because Q is very small near the point of neutrality [12]. In addition, the state equation described above is nonlinear with respect to the flow of base or the control input F_B . Therefore, the dynamic characteristic of the nonlinear equation is sensitive to any change of the control input, and the amount of sampling periods chosen will affect the dynamics of system through the control input F_B .

According to the definition of pH value, we have

$$pH = -\log_{10}[H^+].$$

In chemical equilibrium, the ion-product of water is approximately equal to a constant, i.e.,

$$[H^+][OH^-] = K_w,$$

where $K_w = 10^{-14}$ (mol/l)² is quite accurate at the temperature $25^\circ C$ as quoted by Jutila [10].

So we readily derive the output equation below

$$pH = -\log_{10}(Q + \sqrt{Q^2 + 4K_w}) + \log_{10} 2.$$

Note that there exists the high nonlinearity between a measured pH value and a state variable Q . Fig. 5 gives the curve for the neutralization of strong acid ($C_A = 10^{-3}$ mol/l) with strong base ($C_B = 10^{-3}$ mol/l), where the term F_B/F_A is the ratio of the flow of base F_B to the flow of acid F_A . Observations indicate that near the neutrality point ($pH = 7$), even a small deviation between the flows of acid and base result in great variations in the pH value, whereas the pH value smoothly changes as $pH < 5$ or $pH > 9$. In the pH -neutralization process, such a serious sensitivity of pH values to the control input F_B makes it difficult to establish a sufficiently accurate analytical model. Therefore it is hard to know what the system order should exactly be chosen, if the nonlinear state space representation is expressed as a difference equation called NARMA model.

B. Dataset Preparation

We employed the Runge-Kutta-Fehlberg integration method to solve the differential equation of the pH -

neutralization process. Specifically, we set a stepsize of 0.01 for the integration method and selected the initial state as $Q_0 = 0.0001$. Consequently, one training dataset and two different test datasets were constructed, respectively, by using the *ode45* function in MATLAB when we forced different input signals upon the pH -neutralization process. The sinusoid input signal used for the training dataset was given by $F_B(t) = 0.105 \sin(\pi t/2) + 0.105$ ($0 \leq t \leq 20$), which was strictly restricted to be within the allowable range of control input change specified by practical pH -neutralization process (i.e., $F_B(t) \in [0.000, 0.210]$). As we selected a sampling period of $T_S = 0.02$ (min), a total of 1,001 input-output pairs ($0 \leq t \leq 20$) were produced and then used as our training dataset (Fig. 6(a) and (b)).

To generate test datasets, we used the following two different functions as $F_B(t)$:

$$(a) \text{ Test 1: } y_d(t) = 0.04 \sin(3\pi t/5) \cos(\pi t) + 0.06 \cos(3\pi t/5) + 0.105, \quad 0 \leq t \leq 20.$$

$$(b) \text{ Test 2: } y_d(t) = 0.06 \sin(2\pi t/5) + 0.02 \sin(4\pi t) + 0.02 \cos(\pi t) + 0.105, \quad 0 \leq t \leq 20.$$

Then the pH -value is obtained by MATLAB. As a performance measure, the mean-square error (MSE) was used for evaluating the trained SHESN in our experiment.

C. Simulation Results

Based on the incremental growth model presented above, we first generated a naturally evolving state reservoir of 3,000 internal neurons and then established the whole SHESN network applicable to the pH -neutralization process. The network topology parameters used here were given below: the capacity of reservoir $L \times L = 500 \times 500 = 250,000$, the number of internal neurons $n = 3,000$, the number of backbone neurons $n_b = 30$, and the number of connections for each local neuron $n_c = 10$. In addition, we took the parameters of the bounded Pareto heavy-tailed distribution as follows: $a = 1, k = 1$, and $Q = 250,000$. Accordingly, we produced an $3,000 \times 3,000$ dimensional reservoir matrix W^{res} that had the spectral radius of 0.10 and the sparse connectivity of 0.64%. Furthermore, the noise $v(k)$ forced to the activation function of internal neurons was randomly produced with uniform distribution over $[-0.0001, 0.0001]$. The input and feedback weight matrices, i.e., W^{in} and W^{fb} , were randomly set with uniform distribution over $[-1, 1]$. In particular, the output weight matrix W^{out} , based on the given training dataset, was derived using the one-step learning algorithm. Note that although the SHESN designed here had really a large-scale state reservoir, in which the total number of the internal neurons reached up to 3,000, the learning process can be done only with one step through calculations of a generalized inverse matrix. After the training phase terminated, we began to benchmark the well-trained SHESN using the two test datasets generated. The goal was to examine capabilities of the SHESN in approximating highly nonlinear and quite sensitive dynamic systems. For comparison, we also investigated application of the ESN to the inverse dynamic modeling of the pH -neutralization process. For the ESN, we employed identical

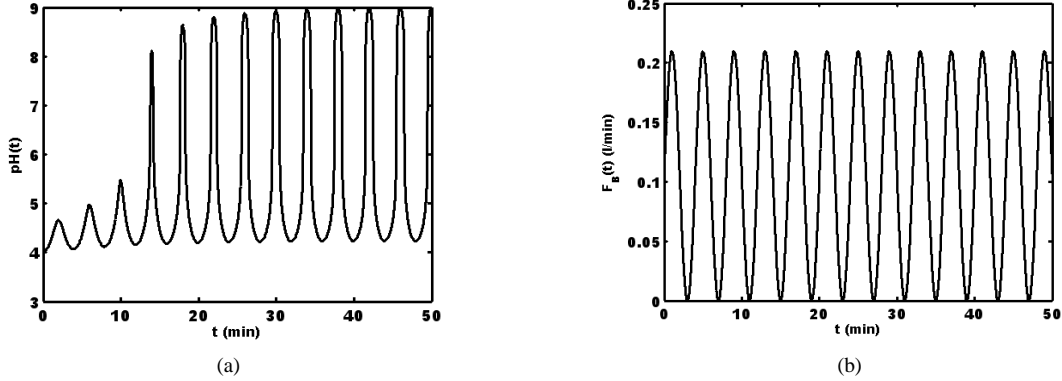


Fig. 6. Input-output pairs of the pH-neutralization process used for constituting one training dataset. (a) pH-value is used as the input of SHESN. (b) Control input F_B as the desired output of SHESN.

training and test datasets generated above and took the same network parameters as the SHESN. Specifically, we set the equal spectral radius of 0.10 for the completely random state reservoir of ESN.

The output errors of both the SHESN and the ESN achieved using the two test datasets are shown in Fig. 7(a) and (b), respectively. In the first test, the MSE test errors of the SHESN and the ESN were calculated to be $2.0828e-008$ and $9.2824e-008$, respectively. In the second test, the MSE test errors were $4.9838e-007$ and $1.6225e-006$, respectively. All the experimental results showed that the proposed SHESN had significantly stronger capabilities of approximating nonlinear dynamic systems than the ESN. In particular, our other comparative study also illustrated that the SHESN outperformed even much better than the previous dynamic fuzzy neural network approach presented in [12], where the MSE test error was surprisingly improved by a factor of about 10^3 to 10^4 .

D. Discussion

Two problems should be treated carefully. The first is the possible over-fitting problem when such a large-scale reservoir is applied. Here we carried out experiments on the SHESNs with reservoir of 500 neurons and 1,000 neurons. Using the first dataset, we had the MSE test errors of $1.4e-6$ and $7.2e-7$, respectively. Based on the second dataset, the resulting MSE test errors were $2.3e-5$ and $3.9e-6$, respectively. Apparently, the SHESN with 3,000 internal neurons outperforms all the two networks with 500 and 1,000 internal neurons. All these results indicates clearly that our SHESN has good generalization capability. The second issue involves the computational efficiency of the proposed model. Actually, both the SHESN and the ESN are very efficient due to the fact that: (a) the training process of networks is only a linear regression problem that can be found by the pseudoinverse function, and (b) the feedforward propagation of either the SHESN or the ESN just needs to do several operations of matrix multiplication and summation in each dynamic step.

IV. NATURAL CHARACTERISTICS AND BIOLOGICAL PLAUSIBILITY

A. Analysis of Natural Characteristics for the SHESN

1) *Small-World Phenomenon: Average Characteristic Path Length and Clustering Coefficient:* The average characteristic path length $L(p)$ and the clustering coefficient $C(p)$ [20] are used to characterize the small-world phenomenon of complex network topology. As mentioned above, our naturally evolving reservoir network was composed of 3,000 internal neurons. Accordingly, we had an $3,000 \times 3,000$ reservoir matrix W^{res} , with a sparse connectivity of 0.64%. Hence it was undoubtedly a large and sparse complex network.

As a global property, the average characteristic path length $L(p)$ reflects the effective size of a complex network. It is defined as the mean distance between two internal neurons, averaged over all pairs of internal neurons. The clustering coefficient $C(p)$ is a local property measuring the probability that two neurons neighboring an internal neuron are also neighbors of each other [20]. More precisely, suppose that the internal neuron v in the new state reservoir has k_v neighbor neurons that are connected to the neuron v . Let $n_a(n_t)$ represent the actual (total possible) number of connections among the neighbor neurons of v . The clustering coefficient C_i of the internal neuron v is defined as $C_i = n_a/n_t = 2n_a/(k_v(k_v - 1))$ [20]. Hence the clustering coefficient $C(p)$ of the whole new reservoir network is the average of C_i over all the internal neurons.

For the SHESN's reservoir presented in this paper, the average characteristic path length and the clustering coefficient were computed as follows: $L(p) = 3.1411$ and $C(p) = 0.4268$, respectively. For comparison, we investigated a completely random reservoir network with the same size of 3,000 internal neurons and a sparse connectivity of 0.8337%. For this ESN reservoir network, we calculated the average characteristic path length and the clustering coefficient, respectively, i.e., $L_{random} = 2.8091$ and $C_{random} = 0.0091$. These calculations indicated that $L(p)$ was almost as small as L_{random} , and $C(p)$ was much greater than C_{random} .

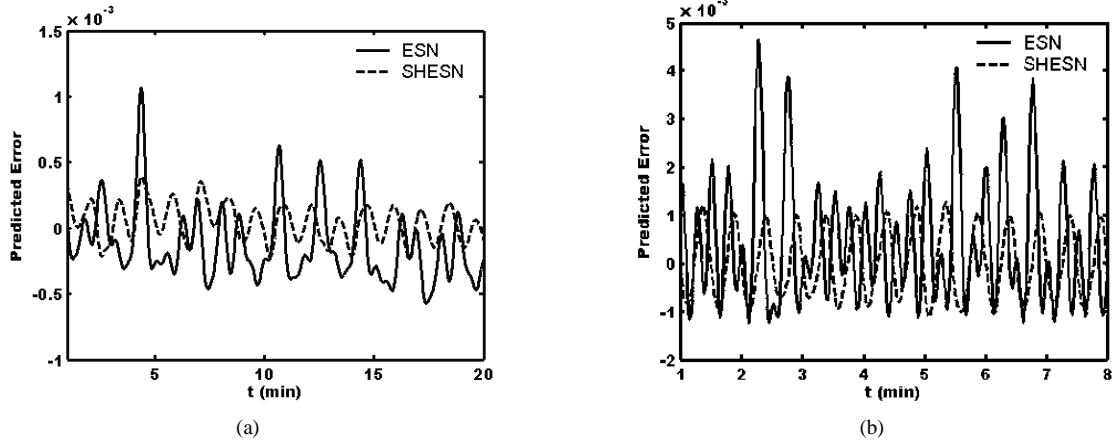


Fig. 7. Predicted errors of ESN and SHESN. (a) Using the first test dataset. (b) Using the second test dataset.

In other words, the new reservoir of the SHESN had a short average characteristic path length and a high clustering coefficient. Therefore our new reservoir is a small-world complex network.

2) *Scale-Free Feature*: It is well known that power laws are free of any characteristic scale. Networks that have power law degree distributions are called scale-free networks. In recent years, empirical studies have unveiled that Internet topologies exhibit power laws in the form of $y = x^{-\alpha}$ [6], [14], [16]. Usually, the power law exponent α is employed to characterize some universal properties of network topology. In order to find the exponent α , we first plotted the relationship between two variables x and y in a log-log scale and the slope of the resulting linear plot was then viewed as the power law exponent α . Furthermore, we utilized Pearson's correlation coefficients for the log-log plot so as to judge whether power laws really exist. In fact, the closer the absolute value of the correlation coefficient is to one, the more closely the data obey a power law [14]. For a good linear fit on a log-log plot, the correlation coefficient should often be greater than 0.95 and the p-value should be less than 0.05 as well. In general, if the p-value is small, then the correlation is significant.

Let us consider the following two scale-free distributions [6]: outdegree of neurons vs. rank and the number of neurons vs. outdegree. Both of them were observed in our naturally evolving reservoir.

Fig. 8 shows the relationship between the outdegree of internal neurons and the rank of neurons on a log-log plot. Using the *corrcoef* function in MATLAB, the correlation coefficient was calculated to be 0.9798 with the p-value of 0. The rank exponent R , or the slope of the fitting linear plot, was calculated as 0.56, which is in accordance with the results for the Internet topology in [6].

Similarly, as shown in Fig. 9, we obtained the relationship between the number of neurons and the outdegree. Note that we eliminated the neurons that have outdegree outliers before linearly fitting the log-log plot. The correlation coefficient was calculated as 0.9746 and the p-value 0. In this case, we

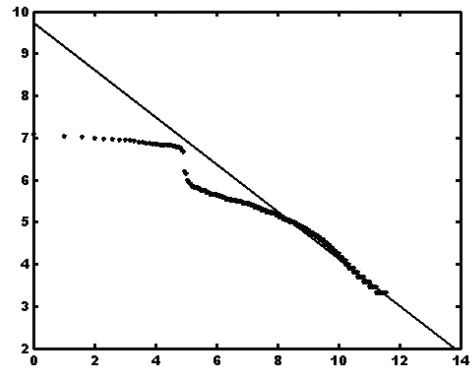


Fig. 8. Log-log plot of outdegree of neurons vs. rank. The correlation coefficient is 0.9798.

computed the outdegree exponent of reservoir $O = 2.50$. Actually, the outdegree exponents of 2.0 – 3.0 have frequently been discovered in most of the natural complex networks such as biological networks [1], [19]. This illustrates that our SHESN expresses some biological characteristics at least in power laws and local preferential attachments.

B. Discussion of Biological Plausibility

In fact, the structure of our natural evolving state reservoir is quite consistent with the functional network of human brain, which has the following features [4]:

- Functional connections obey power-law distribution.
- The characteristic path length is small and comparable with that of equivalent random networks.
- The clustering coefficient is orders of magnitude larger than equivalent random networks.
- The probability of finding a link between two nodes decreases as their distance increases.

As discussed earlier, the network structure of our new state reservoir retains the first three properties. More interestingly, the fourth property implies that both the spatial information for each neuron and the locality of connections, which have been included in our local preferential attachment, are

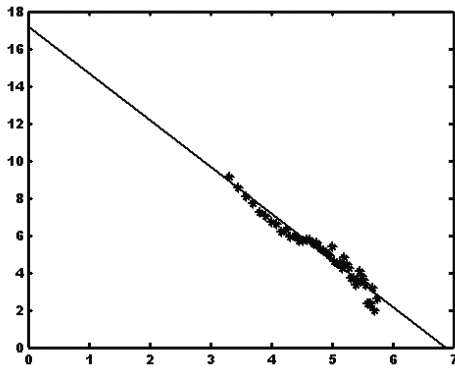


Fig. 9. Log-log plot of the number of neurons vs. outdegree. The correlation coefficient is 0.9746.

essential to producing a topology of small-world and scale-free distribution.

In some sense, the structural characteristics of brain functional network provide the biological plausibility of the SHESN. We expect that such a highly plausible structure plays a critical role in enhancing computational capabilities of the SHESN. This hypothesis is worth being examined in the future.

V. CONCLUSION

In this paper, we proposed a scale-free highly-clustered echo state network (SHESN) for the inverse dynamic modeling of the pH-neutralization process. The SHESN is an extended echo state network that contains a naturally evolving state reservoir. Significantly, the new state reservoir of the SHESN is incrementally generated according to dynamic growth rules, instead of being constructively built, like most of the existing neural network approaches. The design of dynamic growth model took several natural features into account: (1) short average characteristic path length, (2) high clustering coefficient, (3) scale-free distribution for network degree, and (4) spatially hierarchical and distributed structure. We implemented the SHESN that had up to 3,000 internal neurons, and applied it to modeling the PH-neutralization process. Interestingly, neither difference equation nor system order was required to provide in advance for the SHESN-based modeling scheme. Empirical results illustrated that our SHESN was capable of more accurately approximating the inverse dynamic model of the pH-neutralization process than the ESN and other previous traditional approaches in [12]. Moreover, we empirically made analysis of the generated naturally evolving state reservoir, and discussed very interesting natural phenomena present in the new state reservoir. The future work will focus on the relationship between the biologically inspired structure and the enhanced computational capabilities of the SHESN. Research along these lines and practical applications to other complex dynamic systems are in progress.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation of China under Grant 60321002 and the Teaching and Research Award Program for Outstanding Young Teachers in Higher Education Institutions of MOE (TRAPOYT), China.

REFERENCES

- [1] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.
- [2] A.-L. Barabasi and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," *Nature Rev.*, vol. 5, pp. 101–113, 2004.
- [3] M. Crovella, M. Harchol-Balter, and C. Murta, "Task assignment in a distributed system: improving performance by unbalancing load," in *Proc. ACM Sigmetrics '98 Conference on Measurement and Modeling of Computer Systems Poster Session*, Madison, WI, June 1998.
- [4] V. M. Eguiluz, D. R. Chialvo, G. Cecchi, M. Baliki, and A. Apkarian, "Scale-free brain functional networks," *Phys. Rev. Lett.*, 2005.
- [5] J. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–211, 1990.
- [6] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 29, pp. 251–263, 1999.
- [7] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol. 52, pp. 141–152, 1985.
- [8] H. Jaeger, "The echo state approach to analysing and training recurrent neural networks," German National Research Institute for Computer Science, Tech. Rep. GMD-Report 148, Dec. 2001. [Online]. Available: ftp://borneo.gmd.de/pub/indy/publications_herbert/EchoStatesTechRep.pdf
- [9] H. Jaeger and H. Haas, "Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, pp. 78–80, 2004.
- [10] P. Jutila, "Application of adaptive ph-control algorithm," *Int. Journal Control*, vol. 38, pp. 639–655, 1983.
- [11] K. Klemm and V. M. Eguiluz, "Highly clustered scale-free networks," *Phys. Rev. E*, vol. 65, p. 036123, 2002.
- [12] D. P. Kwok and Z. D. Deng, "Dynamic modeling of ph-neutralization process using fuzzy neural network," *Advances in Modeling and Analysis*, vol. 39, pp. 51–66, 1998.
- [13] D. P. Kwok, Z. D. Deng, C. K. Li, T. P. Leung, Z. Q. Sun, and J. K. C. Wong, "Fuzzy neural control of systems with unknown dynamic using q-learning strategies," in *Proc. The IEEE International Conference on Fuzzy Systems (FUZZ- IEEE2003)*, St. Louis, MO, USA, May 2003.
- [14] A. Medina, I. Matta, and J. Byers, "On the origin of power-laws in internet topologies," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 30, pp. 18–28, 2000.
- [15] S. Milgram, "The small world problem," *Psychology Today*, vol. 2, pp. 60–67, 1967.
- [16] Z. L. Ren, Z. D. Deng, D. X. Shuai, and J. Gu, "Analysis of power spectrum and 1/f type power law in a complex computer network model," *Computer Physics Communications*, vol. 136, pp. 225–235, 2001.
- [17] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press, 1986.
- [18] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, pp. 268–276, 2001.
- [19] X. F. Wang and G. R. Chen, "Complex networks: small-world, scale-free and beyond," *IEEE Circuits Syst. Mag.*, vol. 3, pp. 6–20, 2003.
- [20] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.