IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 15, 2018, accepted February 23, 2018, date of publication March 1, 2018, date of current version March 15, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2810190

Adaptive Levenberg-Marquardt Algorithm Based Echo State Network for Chaotic Time Series Prediction

JUNFEI QIAO, LEI WANG, CUILI YANG, AND KE GU
Beijing Key Laboratory of Computational Intelligence and Intelligent System, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

Corresponding author: Junfei Qiao (junfeiq@bjut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61533002 and Grant 61603012, in part by the Beijing Municipal Education Commission Foundation under Grant KM20170005025, in part by the Beijing Post-Doctoral Research Foundation of China under Grant 2017ZZ-028, and in part by the China Post-Doctoral Science Foundation funded Project.

ABSTRACT Echo state networks (ESNs) have wide applications in chaotic time series prediction. In the ESN, if the smallest singular value of the reservoir state matrix is infinitesimal, the ill-posed problem might occur during the training process. To overcome this problem, an adaptive Levenberg–Marquardt (LM) algorithm-based echo state network (ALM-ESN) is developed. In the developed ALM-ESN, a new adaptive damping term is introduced into the LM algorithm. The adaptive factor is amended by the trust region technique, furthermore, convergence analysis, and stability analysis are performed. Moreover, to make the inputs fall within the active region of the activation function and improve the learning speed, a weight initialization method using linear algebra is deployed to determine the appropriate input weights and reservoir weights. Simulations demonstrate that the ALM-ESN can overcome the ill-posed problem. Furthermore, it exhibits better performance and robustness for chaotic time series prediction than some other existing methods.

INDEX TERMS Echo state network, adaptive Levenberg-Marquardt algorithm, trust region technique, weight initialization, chaotic time series prediction.

I. INTRODUCTION

Chaos is a universal phenomenon in nature and human society. Chaos theory can give an appropriate means to demonstrate the properties of dynamic systems [1]. Therefore, research on chaotic time series prediction is very significant [2]–[4]. Many studies have been performed on chaotic time series prediction using neural networks (NNs), such as the fuzzy neural network (FNN) [5], the support vector machine (SVM) [6], the recurrent neural network (RNN) [7] and the echo state network (ESN) [8]. These methods can theoretically approximate dynamic systems with any arbitrary accuracy. Among these models, the ESN has been paid a growing amount of attention. In the ESN, only the weights from the reservoir to the output layer need to be tuned by the least squares method, while the input weights and reservoir weights remain unchanged once generated. Unlike the RNN, the ESN can overcome the local minima and gradient vanishing problems [8], and thus it exhibits better performance than other traditional neural networks. Based on these advantages, the ESN has numerous successful applications such as Sunspot prediction [9], human motion modelling [10], wireless service [11], online learning control [12] and electric load forecasting [13].

Although the ESN has many advantages, some problems still need to be solved. For example, the ill-posed problem might occur during the learning process, which would weaken the generalization ability of the ESN. There are some reasons for this limitation provided below. First, since the input weights and reservoir weights are randomly generated, this random weight initialization can make the inputs fall within the saturation region of the activation function and result in the ill-posed problem. Second, the pseudoinverse serves as the common training method for the ESN [8]. When the smallest singular value of the reservoir state matrix is infinitesimal, the ill-posed problem might happen. To address this problem, the weight initialization method for determining the optimal region of initial weight values and the readout training method for the ESN are considered in this paper.

Generally, the input weights and reservoir weights are important for the learning speed and performance of the ESN [14]. Since the activation function of the reservoir is hyperbolic tangent in the ESN, according to the sensitive area distribution of the hyperbolic tangent function, small weights can make the sensitive area have a certain width. If the input weights and reservoir weights are small, many reservoir neurons will fall within the active region of the activation function. Otherwise, if the scaling is very large, they will stay in the saturation region, resulting in the ill-posed problem. Therefore, it is necessary to determine the appropriate input weights and reservoir weights to carry out weight initialization. Several methods have been developed. Using the independent component analysis, the optimal hidden layer's initial weights are given for the multilayer perceptron and the salient feature components are obtained from the input data [15]. Using the mutual information method, weights are initialized for the FFNN [16]. It is noted that the above-mentioned weight initialization method may have a high computational burden. Thus the developed method based on the Cauchy inequality is given. This developed algorithm is computationally efficient and can ensure the output lies in the active region of the activation function.

To compute the proper output weights in the ESN, some methods have been developed, such as the pseudoinverse solution [8] and the singular value decomposition (SVD) [9]. In these methods, if the smallest singular value of the reservoir state matrix tends to become zero, the state matrix does not have full column rank, resulting in the ill-posed problem. Then, to address this issue, some regularization methods, such as the l_2 penalty (ridge regression [17], also called *Tikhonov* regularization) (RR-ESN) [18], [19] and the l_1 penalty (Lasso-ESN) [20], have been applied to improve the generalization ability. However, the RR-ESN has difficulties in directly obtaining the optimal ridge parameter. Since it is non-convex, some suboptimal solutions will be obtained. To effectively optimize the output layer connection and eliminate unnecessary output layer connection, some evolutionary algorithms, such as the genetic algorithm (GA) [21] and binary particle swarm optimization (BPSO) [22], are employed. However, evolutionary algorithms suffer from high computational complexity and tend to experience premature convergence. According to [23], it is known that the generalization ability degrades due to large output weights. Therefore, it is necessary to solve the large output weights to avoid the occurrence of the ill-posed problem. On the other hand, the LM algorithm is one of the most successful algorithm in increasing the convergence speed and avoiding the occurrence of ill-posed problem [24], [25]. Using the LM algorithm, some satisfactory results have been obtained and good convergence can be ensured. In this paper, the output weights are computed by the LM algorithm to replace the linear regression. Furthermore, a new damping term is adaptively given (called ALM-ESN), where the adaptive factor is amended by the trust region technique.

The rest of this paper is organized as follows. A short review of the classical ESN and an improved ESN based on the LM algorithm is given in Section II. The convergence analysis and stability analysis of the ALM-ESN are shown in Section III and Section IV, respectively. Some experiments are conducted to illustrate the performance of the developed ALM-ESN compared to other existing models in Section V. Some conclusions are presented in Section VI.

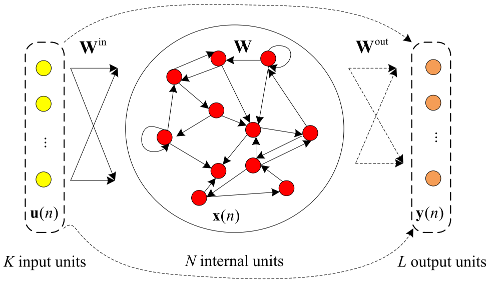

FIGURE 1. The basic architecture of the OESN. The dashed connections are calculated by linear regression.

II. THE DEVELOPED ESN

A. THE ORIGINAL ESN

The original ESN (OESN) [8] has an input layer, reservoir layer and readout layer, as shown in Fig.1 (without feedback connections). The corresponding recursive formula of the OESN is given as follows:

$$\mathbf{x}(n) = \mathbf{g}(\mathbf{W}\mathbf{x}(n-1) + \mathbf{W}^{in}\mathbf{u}(n)), \tag{1}$$

$$\mathbf{y}(n) = \mathbf{W}^{out}(\mathbf{x}(n), \mathbf{u}(n)). \tag{2}$$

where $\mathbf{u}(n) \in \mathbb{R}^K$ denotes the external input, $\mathbf{x}(n) \in \mathbb{R}^N$ represents the reservoir state, $\mathbf{y}(n) \in \mathbb{R}^L$ is the network prediction output and $\mathbf{z}(n) \in \mathbb{R}^L$ is the corresponding desired output. $\mathbf{g}$ is the activation function of the reservoir, which is chosen as the hyperbolic tangent function. The input weight matrix $\mathbf{W}^{in} \in \mathbb{R}^{N\times K}$ and the reservoir weight matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ are randomly generated, and the output weight matrix $\mathbf{W}^{out} \in \mathbb{R}^{L\times(K+N)}$ needs to be calculated using the simple linear regression. The internal state can be provided as $\mathbf{X} = [\mathbf{X}(1), \mathbf{X}(2), \ldots, \mathbf{X}(P)]^T$ (P is the size of training samples), where $\mathbf{X}(n) = [\mathbf{x}(n)^T, \mathbf{u}(n)^T]^T$. The desired output is $\mathbf{Z} = [\mathbf{z}(1), \mathbf{z}(2), \ldots, \mathbf{z}(P)]^T$. For a given training samples set $\left\{(\mathbf{u}(n), \mathbf{z}(n)) | \mathbf{u}(n) \in \mathbb{R}^K, \mathbf{z}(n) \in \mathbb{R}^L\right\}$, the output weights can be calculated as

$$\mathbf{W}^{out} = (\mathbf{X}^{+}\mathbf{Z})^T = ((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Z})^T. \tag{3}$$

where $\mathbf{X}^{+}$ is the Moore-Penrose generalized inverse of the internal state matrix $\mathbf{X}$.

Using the SVD [9], the reservoir state matrix can be decomposed as $\mathbf{X} = \mathbf{A}\mathbf{\Lambda}B^T$, where $\mathbf{A}$ and $\mathbf{B}$ are both orthogonal matrices, $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, \ldots, \lambda_Q)$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_Q > 0$ are the nonzero singular values of $\mathbf{X}$. Then $\mathbf{X}^{+} = \mathbf{B}\mathbf{\Lambda}^{-1}\mathbf{A}^T$, and the corresponding output weights can be rewritten as

$$\mathbf{W}^{out} = (\mathbf{X}^{+}\mathbf{Z})^T = (\mathbf{B}\mathbf{\Lambda}^{-1}\mathbf{A}^T\mathbf{Z})^T = \sum_{i=1}^{Q}\frac{1}{\lambda_i}\mathbf{b}(i)\mathbf{a}(i)^T\mathbf{z}(i) \tag{4}$$

where $\mathbf{a}(i)$ and $\mathbf{b}(i)$ are the i-th column of $\mathbf{A}$ and $\mathbf{B}$, respectively. From formula (4), it is known that if the smallest singular value of $\mathbf{X}$ is close to zero, the output weights are very large, which is ill-posed. As a result, the ESN has bad stability and poor generalization ability.

B. WEIGHT INITIALIZATION

The initial weights have been regarded as one of the most important factors to improve the learning speed and performance of neural networks. In the traditional ESN, the input weight and reservoir weight obey the uniform distribution in the interval $[-1, 1]$ [8]. To test the performance of the network with different weight intervals, the Mackey-Glass system $\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{ax(t-\tau)}{1+x^n(t-\tau)} + bx(t)$ is selected [9]. Fig.2 shows the training ability and generalization ability with different weight interval. Suppose the input weight and reservoir weight obey the uniform distribution in the interval $[-\beta, \beta]$. As shown in Fig.2, when β increases, the mean testing error fluctuates. The mean testing error is at its minimum when β equals 0.3, while the mean training error has a relatively steady change. It means that the weight interval is very important for network performance. Therefore, it is necessary to perform weight initialization to determine the optimal weight interval.

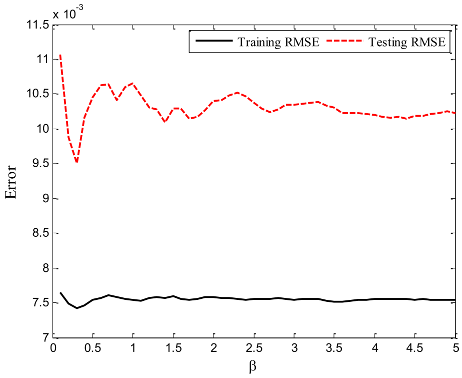

FIGURE 2. **Training and testing error for the Mackey-Glass system.**

Fig.3 shows the curve of hyperbolic tangent function. To ensure that the outputs fall within the active region, the weight should be initialized, resulting in a smaller testing error. The magnitudes of the initial weights are evaluated by the following problem

$$-\bar{s} \le \sum_{j=1}^{N} w_{ij}x_{j,p-1} + \sum_{k=1}^{K} w_{ik}^{in}u_{kp} \le \bar{s} \ (i = 1, \cdots, N), \tag{5}$$

where w_{ik}^{in} is the (i, k)-th element of $\mathbf{W}^{in}$, w_{ij} is the (i, j)-th element of $\mathbf{W}$, $u_{kp}(p = 1, \cdots, P)$ is the k-th element of and $x_{j,p-1}(p = 1, \cdots, P)$ is the j-th element of $\mathbf{x}(p-1)$ $(p = 1, \cdots, P)$.

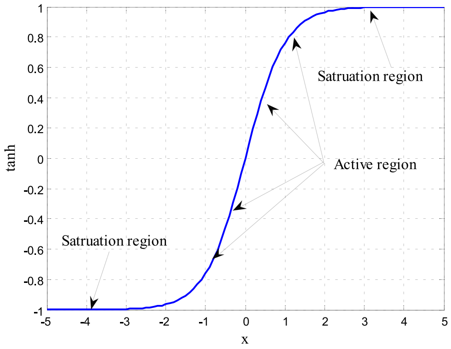

FIGURE 3. **Hyperbolic tangent function.**

In this paper, the active region is assumed to fall within the region, where the derivative of the reservoir activation function is more than 2% of the maximum derivative, i.e., $\bar{s} \approx 2.65$. To determine the optimal weight interval, the following theorem is given.

Theorem 1: The input weights and reservoir weights are supposed to obey the independent uniform distribution with zero mean. If

$$\left(\sum_{k=1}^{K}(u_{kp})^2 + \sum_{j=1}^{N}(x_{j,p-1})^2\right)\left(\sum_{k=1}^{K}(w_{ik}^{in})^2 + \sum_{j=1}^{N}(w_{ij})^2\right) \le \bar{s}^2,$$

the input and reservoir weight interval fall within $[-\beta, \beta]$, where $\beta = \min_{p=1,\cdots,P} \beta_p$,

$$\beta_p \le \bar{s}\sqrt{\frac{3}{(K+N)\left(\sum_{k=1}^{K}(u_{kp})^2 + \sum_{j=1}^{N}(x_{j,p-1})^2\right)}}.$$

Proof: From inequality (5), it can be got that

$$\left(\sum_{j=1}^{N} w_{ij}x_{j,p-1} + \sum_{k=1}^{K} w_{ik}^{in}u_{kp}\right)^2 \le \bar{s}^2,$$

Using the Cauchy inequality,

$$\begin{aligned}&\left(\sum_{j=1}^{N} w_{ij}x_{j,p-1} + \sum_{k=1}^{K} w_{ik}^{in}u_{kp}\right)^2\\ &\le \left(\sum_{k=1}^{K}(u_{kp})^2 + \sum_{j=1}^{N}(x_{j,p-1})^2\right)\left(\sum_{k=1}^{K}(w_{ik}^{in})^2 + \sum_{j=1}^{N}(w_{ij})^2\right)\\ &\le \bar{s}^2.\end{aligned} \tag{6}$$

For the p-th samples, if the input weight and reservoir weight obey the independent uniform distribution with zero mean in the interval $[-\beta_p, \beta_p]$, from the law of large numbers, it can be obtained that

$$\sum_{k=1}^{K}(w_{ik}^{in})^2 \approx E(\sum_{k=1}^{K}(w_{ik}^{in})^2) = K \cdot \mathrm{var}(w_{ik}^{in}) = \frac{K\beta_p^2}{3}. \tag{7}$$

Similarly,

$$\sum_{j=1}^{N}(w_{ij})^2 \approx \frac{N\beta_p^2}{3}. \tag{8}$$

Then

$$\beta_p \leq \bar{s}\sqrt{\frac{3}{(K+N)\left(\sum\limits_{k=1}^{K}(u_{kp})^2+\sum\limits_{j=1}^{N}(x_{j,p-1})^2\right)}}, \tag{9}$$

Let

$$\beta = \min_{p=1,\cdots,P}\beta_p. \tag{10}$$

C. ALM-ESN

The LM algorithm can combine the advantages of the steepest descent method and Gauss-Newton method [25]. It not only possesses the speed advantage of Gauss-Newton method but also has the stability of the steepest descent method. Using the approximate second-order derivative, the LM algorithm converges much faster than the first-order gradient method.

The calculation of the output weights $\mathbf{W}^{out}$ based on the LM algorithm is equivalent to minimizing the objective function $E(\mathbf{W}^{out})$, which can be defined as follows:

$$E(\mathbf{W}^{out}) = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{M}(y_j^p - d_j^p)^2 = \frac{1}{2}\sum_{q=1}^{Q}e_q^2, \tag{11}$$

where $e_q = y_j^q - d_j^q$, y_j^q is the desired output, and d_j^q is the network output.

During each iteration, the output weights $\mathbf{W}^{out}$ will be replaced by the new one. The update rule based on the LM algorithm can be written as follows:

$$\mathbf{W}^{out}(k+1) = \mathbf{W}^{out}(k) - (\mathbf{J}_k^T\mathbf{J}_k + \mu_k\mathbf{I})^{-1}\mathbf{J}_k^T\mathbf{e}_k, \tag{12}$$

where $\mathbf{J}_k$ is a Jacobi matrix, $\mathbf{e}_k$ is an error vector, and μ_k is a positive damping term.

$$\mathbf{e}_k \triangleq \mathbf{e}(\mathbf{W}^{out}(k)) = (e_1, e_2, \cdots, e_Q)^T, \tag{13}$$

$$\mathbf{J}_k \triangleq \mathbf{J}(\mathbf{W}^{out}(k)) = \begin{pmatrix} \frac{\partial e_1}{\partial w_1^{out}} & \frac{\partial e_1}{\partial w_2^{out}} & \cdots & \frac{\partial e_1}{\partial w_{K+N}^{out}} \\ \frac{\partial e_2}{\partial w_1^{out}} & \frac{\partial e_2}{\partial w_2^{out}} & \cdots & \frac{\partial e_2}{\partial w_{K+N}^{out}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_Q}{\partial w_1^{out}} & \frac{\partial e_Q}{\partial w_2^{out}} & \cdots & \frac{\partial e_Q}{\partial w_{K+N}^{out}} \end{pmatrix}. \tag{14}$$

According to (12), since μ_k is a positive damping term, $\mathbf{J}_k^T\mathbf{J}_k + \mu_k\mathbf{I}$ is nonsingular, which can avoid the ill-posed problem. There are many choices for the damping term μ_k. However, there is no general rule in the selecting method of the damping term. In [26], the parameter is chosen as $\mu_k = \|\mathbf{e}_k\|^2$, and it can be shown that the LM algorithm possesses quadratic convergence. However, the damping term $\mu_k = \|\mathbf{e}_k\|^2$ (if no other specified, the operator $\|\cdot\|$ refers to standard l_2 norm in this paper) has some drawbacks. If the sequence trends towards the solution set, $\mu_k = \|\mathbf{e}_k\|^2$ may be less than the machine accuracy. Therefore, it may have no effect. Moreover, $\mu_k = \|\mathbf{e}_k\|^2$ may be very large when the sequence deviates from the solution set, and the step $\mathbf{d}_k$ will approach zero. Consequently, the iteration speed has no advantage. In [27], the parameter is chosen as $\mu_k = \theta\,\|\mathbf{e}_k\| + (1-\theta)\left\|\mathbf{J}_k^T\mathbf{e}_k\right\|$ ($\theta \in [0,1]$) and has a local error bound. It has been shown that the sequence converges quadratically, however, the global convergence is not considered.

Based on these observations, to obtain an appropriate iteration step $\mathbf{d}_k$ and increase the convergence speed, a new damping term is chosen as $\mu_k = \alpha_k\left\|\mathbf{J}_k^T\mathbf{e}_k\right\|^{\delta}$ with $\delta \in [1,2]$, where α_k is an adaptive factor. The trust region technique is used to ensure the global convergence. The actual reduction and predictive reduction of the objective function can be defined as follows:

$$Are\mathbf{d}_k = \|\mathbf{e}_k\|^2 - \left\|\mathbf{e}(\mathbf{W}^{out}(k)+\mathbf{d}_k)\right\|^2, \tag{15}$$

$$Pred_k = \|\mathbf{e}_k\|^2 - \|\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k\|^2. \tag{16}$$

The ratio $r_k = \frac{Are\mathbf{d}_k}{Pre\mathbf{d}_k}$ is important to adopt the trial step and update the parameter α_k between these reductions.

Based on the LM algorithm, each iteration may be described as follows.

$$\begin{cases} \mathbf{W}^{out}(k+1) = \mathbf{W}^{out}(k) + \mathbf{d}(k), \\ \mathbf{d}_k = -(\mathbf{J}_k^T\mathbf{J}_k + \mu_k\mathbf{I})^{-1}\mathbf{J}_k^T\mathbf{e}_k, \\ \mu_k = \alpha_k\left\|\mathbf{J}_k^T\mathbf{e}_k\right\|^{\delta}, \quad \delta \in [1,2]. \end{cases} \tag{17}$$

The main steps of the ALM-ESN can be summarized as follows.

Algorithm 2:

Step 1: Determine the input weight and reservoir weight interval $[-\beta, \beta]$ using inequality (9) and equation (10).

Step 2: Randomly create a reservoir weight matrix $\mathbf{W}_0$ with the given sparsity and reservoir size in the interval $[-\beta, \beta]$. Scale $\mathbf{W}_0$ to $\mathbf{W} = (\alpha_{\mathbf{W}}/\rho(\mathbf{W}_0))\mathbf{W}_0$, where $0 < \alpha_{\mathbf{W}} < 1$ and $\rho(\mathbf{W}_0)$ is the spectral radius of $\mathbf{W}_0$. Initialize the internal state $\mathbf{x}(0)$.

Step 3: Randomly produce an input weight matrix $\mathbf{W}^{in}$ according to a uniform distribution in the interval $[-\beta, \beta]$, initialize the output matrix $\mathbf{W}^{out}(0)$.

Step 4: Obtain the internal states by the external input as (1) from the initial transient $n_{\min}$.

Step 5: Compute the network output, the error vector $\mathbf{e}_k$, the objective function $E(\mathbf{W}^{out})$ and the Jacobi matrix $\mathbf{J}_k$.

Step 6: Given $\varepsilon \geq 0, \alpha_1 > m > 0, 0 \leq p_0 \leq p_1 \leq p_2 < 1$, if the norm of energy function's gradient $\left\|\mathbf{J}_k^T\mathbf{e}_k\right\| \leq \varepsilon$, stop; otherwise compute $\mathbf{d}_k = -(\mathbf{J}_k^T\mathbf{J}_k + \mu_k\mathbf{I})^{-1}\mathbf{J}_k^T\mathbf{e}_k$, where $\mu_k = \alpha_k\left\|\mathbf{J}_k^T\mathbf{e}_k\right\|^{\delta}, \delta \in [1,2], \varepsilon = 10^{-6}, \alpha_1 = 10^{-7}, m = 10^{-8}, p_0 = 0.0001, p_1 = 0.25, p_2 = 0.75.$

Step 7: Compute $r_k = Are\mathbf{d}_k / Pre\mathbf{d}_k$, let

$$\mathbf{W}^{out}(k+1) = \begin{cases} \mathbf{W}^{out}(k) + \mathbf{d}_k, & if\ r_k > p_0, \\ \mathbf{W}^{out}(k), & otherwise. \end{cases}$$

Step 8: Compute

$$\alpha_{k+1} = \begin{cases} 4\alpha_k, & if\ r_k < p_1, \\ \alpha_k, & if\ r_k \in [p_1, p_2], \\ \max\{\frac{\alpha_k}{4}, m\}, & if\ r_k > p_2. \end{cases}$$

go to Step 5.

Step 9: Test the trained ALM-ESN.

III. CONVERGENCE ANALYSIS

Let

$$e(\mathbf{W}^{out}) = 0. \tag{18}$$

Suppose the solution of (18) is nonempty and denote by Ω. It is obvious that

$$\mathbf{d}_k = -(\mathbf{J}_k^T \mathbf{J}_k + \mu_k \mathbf{I})^{-1} \mathbf{J}_k^T \mathbf{e}_k, \tag{19}$$

is a solution of

$$\min_{\mathbf{d}} \theta^k(\mathbf{d}) = \|\mathbf{J}_k \mathbf{d} + \mathbf{e}_k\|^2 + \mu_k \|\mathbf{d}\|^2 . \tag{20}$$

Define

$$\Delta_k = \left\| (\mathbf{J}_k^T \mathbf{J}_k + \mu_k \mathbf{I})^{-1} \mathbf{J}_k^T \mathbf{e}_k \right\| . \tag{21}$$

It can be determined that (12) is equivalent to the following trust region problem:

$$\begin{aligned} &\min_{d} \ \|\mathbf{J}_k \mathbf{d} + \mathbf{e}_k\|^2 \\ &s.t. \ \|\mathbf{d}\| \leq \Delta_k. \end{aligned} \tag{22}$$

Therefore, the LM algorithm is equivalent to the trust region method. To study the convergence properties of the algorithm, we suppose that the following two assumptions and lemma are satisfied.

Assumption 3: $\mathbf{e}_k$ is continuously differentiable. Both $\mathbf{e}_k$ and its Jacobi matrix $\mathbf{J}_k$ are Lipschitz continuous, i.e., there exist positive constants L_1 and L_2 such that

(a) $\left\|\mathbf{J}(\mathbf{W}_1^{out}) - \mathbf{J}(\mathbf{W}_2^{out})\right\| \leq L_1 \left\|\mathbf{W}_1^{out} - \mathbf{W}_2^{out}\right\|$, $\forall \mathbf{W}_1^{out}, \mathbf{W}_2^{out}$;

(b) $\left\|\mathbf{e}(\mathbf{W}_1^{out}) - \mathbf{e}(\mathbf{W}_2^{out})\right\| \leq L_2 \left\|\mathbf{W}_1^{out} - \mathbf{W}_2^{out}\right\|$, $\forall \mathbf{W}_1^{out}, \mathbf{W}_2^{out}$.

By Assumption 3, it can be obtained

$$\begin{aligned} &\left\|\mathbf{e}(\mathbf{W}_1^{out}) - \mathbf{e}(\mathbf{W}_2^{out}) - \mathbf{J}(\mathbf{W}_1^{out})(\mathbf{W}_1^{out} - \mathbf{W}_2^{out})\right\| \\ &\qquad \leq L_1 \left\|\mathbf{W}_1^{out} - \mathbf{W}_2^{out}\right\|, \quad \forall \mathbf{W}_1^{out}, \ \mathbf{W}_2^{out}. \end{aligned}$$

Assumption 4: $\left\|e(\mathbf{W}^{out})\right\|$ provides a local error bound on $N(\mathbf{W}_*^{out}, b_1)$ for (17), i.e., there exist two constants $c_1 > 0$ and $b_1 < 1$ such that

$$\left\|e(\mathbf{W}^{out})\right\| \geq c_1 dist(\mathbf{W}^{out}, \Omega), \quad \forall \mathbf{W}^{out} \in \Omega,$$

where $dist(\mathbf{W}^{out}, \Omega) = \min_{\hat{\mathbf{W}}^{out} \in \Omega} \left\|\mathbf{W}^{out} - \hat{\mathbf{W}}^{out}\right\|$, $N(\mathbf{W}_*^{out}, b_1) = \left\{\mathbf{W}^{out} \middle| \left\|\mathbf{W}^{out} - \mathbf{W}_*^{out}\right\| \leq b_1\right\}, \mathbf{W}_*^{out} \in \Omega$.

Lemma 5 [28]: Let $\mathbf{d}_k$ be computed by Algorithm 2. Then the predicted reduction satisfies

$$Pre\mathbf{d}_k \geq \left\|\mathbf{J}_k^T \mathbf{e}_k\right\| \min\left\{\|\mathbf{d}_k\|, \left\|\mathbf{J}_k^T \mathbf{e}_k\right\| / \left\|\mathbf{J}_k^T \mathbf{J}_k\right\|\right\}.$$

To show the global convergence of Algorithm 2, the following theorem is given.

Theorem 6: The sequence $\left\{\mathbf{W}^{out}(k)\right\}$ generated by Algorithm 2 satisfies $\lim_{k \to \infty} \left\|\mathbf{J}_k^T \mathbf{e}_k\right\| = 0$.

Proof: If the theorem is not true, then there exists a constant $\tau > 0$ and infinitely many k such that $\|\mathbf{J}_k \mathbf{e}_k\| \geq \tau$.

Let

$$\begin{aligned} K &= \left\{k \middle| \left\|\mathbf{J}_k^T \mathbf{e}_k\right\| \geq \tau \right\}, \\ T &= \left\{k \middle| \mathbf{W}^{out}(k+1) \neq \mathbf{W}^{out}(k), k \in K \right\}. \end{aligned}$$

Using Lemma 5, it can be got that

$$\begin{aligned} \|\mathbf{e}_1\|^2 &\geq \sum_{k \in K} (\|\mathbf{e}_k\|^2 - \|\mathbf{e}_{k+1}\|^2) = \sum_{k \in T} (\|\mathbf{e}_k\|^2 - \|\mathbf{e}_{k+1}\|^2) \\ &\geq \sum_{k \in T} p_0 Pre\mathbf{d}_k \geq \sum_{k \in T} p_0 \left\|\mathbf{J}_k^T \mathbf{e}_k\right\| \\ &\quad \times \min\{\|\mathbf{d}_k\|, \left\|\mathbf{J}_k^T \mathbf{e}_k\right\| / \left\|\mathbf{J}_k^T \mathbf{J}_k\right\|\}. \end{aligned}$$

Using Algorithm 2, it can be obtained that $\|\mathbf{d}_k\| = \left\|(\mathbf{J}_k^T \mathbf{J}_k + \mu_k \mathbf{I})^{-1}\right\| \left\|\mathbf{J}_k^T \mathbf{e}_k\right\| \leq \left\|\mathbf{J}_k^T \mathbf{e}_k\right\| / \left\|\mathbf{J}_k^T \mathbf{J}_k\right\|$. Hence $\min\{\|\mathbf{d}_k\|, \left\|\mathbf{J}_k^T \mathbf{e}_k\right\| / \left\|\mathbf{J}_k^T \mathbf{J}_k\right\|\} = \|\mathbf{d}_k\|$, which implies that $\sum_{k \in T} \|\mathbf{d}_k\| < +\infty$.

Since there are infinitely many k satisfying $\|\mathbf{J}_k \mathbf{e}_k\| \geq \tau$, there exists $\hat{k}$, such that $\|\mathbf{J}_k \mathbf{e}_k\| \geq \tau$ and $\sum_{k \geq \hat{k}} \|\mathbf{d}_k\| < +\infty$ for all $k \geq \hat{k}$. This result implies that $\lim_{k \to \infty} \mathbf{W}^{out}(k)$ exists, which shows that $\alpha_k \to +\infty$.

Additionally, it follows from $\|\mathbf{J}_k \mathbf{e}_k\| \geq \tau$, $\sum_{k \geq \hat{k}} \|\mathbf{d}_k\| < +\infty$ for all $k \geq \hat{k}$ and Lemma 5 that

$$\begin{aligned} r_k &= \frac{Are\mathbf{d}_k}{Pre\mathbf{d}_k} = 1 + \frac{\|\mathbf{e}_k + \mathbf{J}_k \mathbf{d}_k\| \, O(\|\mathbf{d}_k\|^2) + O(\|\mathbf{d}_k\|^4)}{Pre\mathbf{d}_k} \\ &\leq 1 + \frac{\|\mathbf{e}_k + \mathbf{J}_k \mathbf{d}_k\| \, O(\|\mathbf{d}_k\|^2) + O(\|\mathbf{d}_k\|^4)}{\left\|\mathbf{J}_k^T \mathbf{e}_k\right\| \min\left\{\|\mathbf{d}_k\|, \left\|\mathbf{J}_k^T \mathbf{e}_k\right\| / \left\|\mathbf{J}_k^T \mathbf{J}_k\right\|\right\}} \\ &\leq 1 + \frac{O(\|\mathbf{d}_k\|^2)}{\|\mathbf{d}_k\|} \to 1. \end{aligned}$$

Based on Algorithm 2, there is a constant $M > 0$ satisfying $\alpha_k < M$ for all large k, which is contradictory. □

Before discussing the local convergence, the following two lemmas are introduced. Suppose $\left\{\mathbf{W}^{out}(k)\right\}$ is sufficiently close to Ω, i.e., $dist(\mathbf{W}^{out}(k), \Omega) \ll 1$. Let $\bar{\mathbf{W}}^{out}(k) \in \Omega$ satisfy $\left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\| = dist(\mathbf{W}^{out}, \Omega)$.

Lemma 7 [27]: If $\mathbf{W}^{out}(k) \in N(\mathbf{W}_*^{out}, b_1)$, then there is a constant $c_2 > 0$ satisfying

$$\begin{aligned} c_2 dist(\mathbf{W}^{out}, \Omega)^{\delta} &\leq \mu_k \\ &= \alpha_k \left\|\mathbf{J}_k^T \mathbf{e}_k\right\|^{\delta} \leq L_3 \left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\|^{\delta}. \end{aligned}$$

Lemma 8 [27]: $\|\mathbf{d}_k\| \leq O(\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\|)$.

Using the SVD of the Jacobi matrix, the quadratic convergence of Algorithm 2 is studied. Suppose that the SVD of $\mathbf{J}(\mathbf{W}_*^{out})$ is

$$\begin{aligned}\mathbf{J}(\mathbf{W}_*^{out}) &= \mathbf{U}^*\mathbf{\Sigma}^*\mathbf{V}*^T = (\mathbf{U}_1^*, \mathbf{U}_2^*)\begin{pmatrix}\mathbf{\Sigma}_1^* & \\ & \mathbf{O}\end{pmatrix}\begin{pmatrix}\mathbf{V}_1^{*T} \\ \mathbf{V}_2^{*T}\end{pmatrix} \\ &= \mathbf{U}_1^*\mathbf{\Sigma}_1^*\mathbf{V}_1^{*T}.\end{aligned}$$

where $\Sigma_1^* = diag(\sigma_1^*, \cdots, \sigma_r^*), \sigma_1^* \geq \sigma_2^* \geq \ldots \geq \sigma_r^* > 0, rank(\mathbf{\Sigma}_1^*) = r$.

Suppose that the SVD of $\mathbf{J}(\mathbf{W}^{out}(k)) \triangleq \mathbf{J}_k$ is as follows.

$$\begin{aligned}\mathbf{J}_k &= \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T \\ &= (\mathbf{U}_{k,1}, \mathbf{U}_{k,2}, \mathbf{U}_{k,3})\begin{pmatrix}\mathbf{\Sigma}_{k,1} & & \\ & \mathbf{\Sigma}_{k,2} & \\ & & O\end{pmatrix}\begin{pmatrix}\mathbf{V}_{k,1}^T \\ \mathbf{V}_{k,2}^T \\ \mathbf{V}_{k,3}^T\end{pmatrix} \\ &= \mathbf{U}_{k,1}\mathbf{\Sigma}_{k,1}\mathbf{V}_{k,1}^T + \mathbf{U}_{k,2}\mathbf{\Sigma}_{k,2}\mathbf{V}_{k,2}^T,\end{aligned}$$

where $\Sigma_{k,1} = diag(\sigma_1^{(k)}, \ldots, \sigma_r^{(k)})$, $\Sigma_{k,2} = diag(\sigma_{r+1}^{(k)}, \ldots, \sigma_{r+q}^{(k)})$, $\sigma_1^{(k)} \geq \ldots \geq \sigma_r^{(k)} \geq \sigma_{r+1}^{(k)} \geq \ldots \geq \sigma_{r+q}^{(k)} > 0, q \geq 0$.

For convenience, denote $\mathbf{\Sigma}_{k,i}$, $\mathbf{U}_{k,i}$ and $\mathbf{V}_{k,i}$ as $\mathbf{\Sigma}_i$, $\mathbf{U}_i$ and $\mathbf{V}_i(i = 1, 2, 3)$, respectively. Consequently, the SVD of $\mathbf{J}_k$ can be written as $\mathbf{J}_k = \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^T + \mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2^T$.

Lemma 9 [29]:

(a) $\left\|\mathbf{U}_1\mathbf{U}_1^T\mathbf{e}_k\right\| \leq O(\left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\|)$;

(b) $\left\|\mathbf{U}_2\mathbf{U}_2^T\mathbf{e}_k\right\| \leq O(\left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}(k)\right\|)$;

(c) $\left\|\mathbf{U}_3\mathbf{U}_3^T\mathbf{e}_k\right\| \leq O(\left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\|^2)$.

The local convergence of Algorithm 2 is given as follows.

Theorem 10: The sequence $\left\{\mathbf{W}^{out}(k)\right\}$ created by Algorithm 2 quadratically converges to the solution of (18).

Proof: First, it will be proven that $r_k \to 1(k \to \infty)$.

We consider the following two aspects:

1) If $\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\| \leq \mathbf{d}_k$, from Lemma 8, it can be got that

$$\begin{aligned}&\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\| = O(\|\mathbf{d}_k\|), \\ &\|\mathbf{e}_k\| - \|\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k\| \\ &\quad \geq \|\mathbf{e}_k\| - \left\|\mathbf{e}_k + \mathbf{J}_k(\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k))\right\| \\ &\quad \geq c_1\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\| - L_1\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\| \\ &\quad \geq \hat{c}_1(\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\|) = O(\|\mathbf{d}_k\|).\end{aligned}$$

2) If $\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\| > \mathbf{d}_k$,

$$\begin{aligned}&\|\mathbf{e}_k\| - \|\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k\| \\ &\quad \geq \|\mathbf{e}_k\| \\ &\qquad - \left\|\mathbf{e}_k + \frac{\|\mathbf{d}_k\|}{\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\|}\mathbf{J}_k(\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k))\right\| \\ &\quad \geq \frac{\|\mathbf{d}_k\|}{\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\|} \\ &\qquad \times (\|\mathbf{e}_k\| - \left\|\mathbf{e}_k + \mathbf{J}_k(\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k))\right\|) \\ &\quad \geq \frac{\|\mathbf{d}_k\|}{\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\|}(c_1\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\| \\ &\qquad + O(\left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\|^2)) \\ &\quad \geq \bar{c}_1\|\mathbf{d}_k\|,\end{aligned}$$

$$\begin{aligned}&Pre\mathbf{d}_k \\ &\quad = (\|\mathbf{e}_k\| + \|\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k\|)(\|\mathbf{e}_k\| - \|\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k\|) \\ &\quad \geq \|\mathbf{e}_k\|(\|\mathbf{e}_k\| - \|\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k\|) \geq \|\mathbf{e}_k\| O(\|\mathbf{d}_k\|), \\ &r_k = \frac{Are\mathbf{d}_k}{Pre\mathbf{d}_k} = 1 + \frac{\|\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k\| O(\|\mathbf{d}_k\|^2) + O(\|\mathbf{d}_k\|^4)}{Pre\mathbf{d}_k} \\ &\quad \leq 1 + \frac{\|\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k\| O(\|\mathbf{d}_k\|^2) + O(\|\mathbf{d}_k\|^4)}{\left\|\mathbf{J}_k^T\mathbf{e}_k\right\| \min\left\{\|\mathbf{d}_k\|, \left\|\mathbf{J}_k^T\mathbf{e}_k\right\| / \left\|\mathbf{J}_k^T\mathbf{J}_k\right\|\right\}} \\ &\quad \leq 1 + \frac{O(\|\mathbf{d}_k\|^2)}{\|\mathbf{d}_k\|} \to 1.\end{aligned}$$

There is a constant $M > m$ satisfying $\alpha_k < M$ for all large k.

Second, the quadratic convergence will be proven. Using the SVD of $\mathbf{J}_k$, the following holds,

$$\begin{aligned}&(\mathbf{J}_k^T\mathbf{J}_k + \mu_k\mathbf{I})^{-1} \\ &\quad = \mathbf{V}_1(\mathbf{\Sigma}_1^2 + \mu_k\mathbf{I})^{-1}\mathbf{V}_1^T + \mathbf{V}_2(\mathbf{\Sigma}_2^2 + \mu_k\mathbf{I})^{-1}\mathbf{V}_2^T. \\ &\mathbf{d}_k \\ &\quad = -(\mathbf{J}_k^T\mathbf{J}_k + \mu_k\mathbf{I})^{-1}\mathbf{J}_k^T\mathbf{e}_k \\ &\quad = -\mathbf{V}_1(\mathbf{\Sigma}_1^2 + \mu_k\mathbf{I})^{-1}\mathbf{\Sigma}_1\mathbf{U}_1^T\mathbf{e}_k - \mathbf{V}_2(\mathbf{\Sigma}_2^2 + \mu_k\mathbf{I})^{-1}\Sigma_2\mathbf{U}_2^T\mathbf{e}_k. \\ &\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k \\ &\quad = (\mathbf{U}_1\mathbf{U}_1^T + \mathbf{U}_2\mathbf{U}_2^T + \mathbf{U}_3\mathbf{U}_3^T)\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k \\ &\quad = (\mathbf{U}_1\mathbf{U}_1^T + \mathbf{U}_2\mathbf{U}_2^T + \mathbf{U}_3\mathbf{U}_3^T)\mathbf{e}_k \\ &\qquad - \mathbf{U}_1\mathbf{\Sigma}_1(\mathbf{\Sigma}_1^2 + \mu_k\mathbf{I})^{-1}\mathbf{\Sigma}_1\mathbf{U}_1^T\mathbf{e}_k \\ &\qquad - \mathbf{U}_2\mathbf{\Sigma}_2(\mathbf{\Sigma}_2^2 + \mu_k\mathbf{I})^{-1}\mathbf{\Sigma}_2\mathbf{U}_2^T\mathbf{e}_k \\ &\quad = (\mathbf{U}_1\mathbf{U}_1^T - \mathbf{U}_1\mathbf{\Sigma}_1(\mathbf{\Sigma}_1^2 + \mu_k\mathbf{I})^{-1}\mathbf{\Sigma}_1\mathbf{U}_1^T)\mathbf{e}_k \\ &\qquad + (\mathbf{U}_2\mathbf{U}_2^T - \mathbf{U}_2\mathbf{\Sigma}_2(\mathbf{\Sigma}_2^2 + \mu_k\mathbf{I})^{-1}\mathbf{\Sigma}_2\mathbf{U}_2^T)\mathbf{e}_k + \mathbf{U}_3\mathbf{U}_3^T\mathbf{e}_k \\ &\quad = \mathbf{U}_1(\mathbf{I} - \mathbf{\Sigma}_1(\mathbf{\Sigma}_1^2 + \mu_k\mathbf{I})^{-1}\mathbf{\Sigma}_1)\mathbf{U}_1^T\mathbf{e}_k \\ &\qquad + \mathbf{U}_2(\mathbf{I} - \mathbf{\Sigma}_2(\mathbf{\Sigma}_2^2 + \mu_k\mathbf{I})^{-1}\mathbf{\Sigma}_2)\mathbf{U}_2^T\mathbf{e}_k + \mathbf{U}_3\mathbf{U}_3^T\mathbf{e}_k \\ &\quad = \mu_k\mathbf{U}_1(\mathbf{\Sigma}_1^2 + \mu_k\mathbf{I})^{-1}\mathbf{U}_1^T\mathbf{e}_k + \mu_k\mathbf{U}_2(\mathbf{\Sigma}_2^2 + \mu_k\mathbf{I})^{-1}\mathbf{U}_2^T\mathbf{e}_k \\ &\qquad + \mathbf{U}_3\mathbf{U}_3^T\mathbf{e}_k.\end{aligned}$$

Since $\left\{\mathbf{W}^{out}(k)\right\}$ converges to $\mathbf{W}_*^{out}$, assume that $L_1\left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}\right\| \leq \frac{\sigma_r^*}{2}$.

$$\begin{aligned}&\left\|(\mathbf{\Sigma}_1^2 + \mu_k\mathbf{I})^{-1}\right\| \\ &\quad \leq \left\|\mathbf{\Sigma}_1^{-2}\right\| \leq \frac{1}{(\sigma_r^* - L_1\left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}\right\|)} < \frac{4}{\sigma_r^{*2}}, \\ &\left\|(\mathbf{\Sigma}_2^2 + \mu_k\mathbf{I})^{-1}\right\| \leq \mu_k^{-1}. \\ &\|\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k\| \\ &\quad \leq O(\left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\|^{1+\delta}) \\ &\qquad + O(\left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}\right\|^2) \\ &\quad \leq O(\left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}\right\|^2), \\ &c_1 dist(\mathbf{W}^{out}(k+1), \Omega) \\ &\quad \leq \|\mathbf{e}_{k+1}\| = \|\mathbf{e}_{k+1} + \mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k - \mathbf{e}_k - \mathbf{J}_k\mathbf{d}_k\| \\ &\quad \leq \|\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k\| + \|\mathbf{e}_{k+1} - \mathbf{e}_k - \mathbf{J}_k\mathbf{d}_k\| \\ &\quad \leq \|\mathbf{e}_k + \mathbf{J}_k\mathbf{d}_k\| + O(\|\mathbf{d}_k\|^2) \\ &\quad \leq O(\left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}\right\|^2) \leq O(\left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\|).\end{aligned}$$

It follows from $\left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\| \leq \|\mathbf{d}_k\| + \left\|\mathbf{W}^{out}(k+1) - \bar{\mathbf{W}}^{out}(k+1)\right\|$ that $\left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\| \leq 2\|\mathbf{d}_k\|$ holds for all sufficiently large k.

$\|\mathbf{d}_{k+1}\| = O(\|\mathbf{d}_k\|^2)$, which implies that $\left\{\mathbf{W}^{out}(k)\right\}$ converges quadratically to $\mathbf{W}^{out}_*$, namely

$$\left\|\mathbf{W}^{out}(k+1) - \mathbf{W}^{out}_*\right\| = O(\left\|\mathbf{W}^{out}(k) - \mathbf{W}^{out}_*\right\|^2).$$

This completes the proof. □

IV. STABILITY ANALYSIS

The core of the ESN is that the echo state property (ESP) should be possessed for the reservoir. In other words, the internal states should uniquely depend on the external input. Generally, the ESP is related to the reservoir weight matrix and the input samples. To illustrate the ESP, consider the local dynamics of the system by linearizing the ALM-ESN. The nonlinear system (1) can be approximated as follows.

$$\mathbf{x}(n) = \mathbf{g}'\ \mathbf{W}\mathbf{x}(n-1) + \mathbf{g}'\mathbf{W}^{in}\mathbf{u}(n) \triangleq \mathbf{A}\mathbf{x}(n-1) + \mathbf{B}\mathbf{u}(n) \tag{23}$$

where $\mathbf{g}' = \mathbf{tanh}'$ is the derivative of $\mathbf{tanh}$, $\left\|\mathbf{g}'\right\| \leq 1$, and $\mathbf{A} = \mathbf{g}'\ \mathbf{W}, \mathbf{B} = \mathbf{g}'\ \mathbf{W}^{in}$.

The existence of the ESP may be verified in terms of the necessary condition and sufficient condition of the reservoir matrix [8]. The sufficient condition is that the maximal singular value of $\mathbf{W}$ is less than 1 ($\sigma(\mathbf{W}) < 1$). Since $\|\mathbf{W}\| = \sigma(\mathbf{W})$, the sufficient condition is equivalent to $c \triangleq \|\mathbf{W}\| < 1$.

Suppose that $\mathbf{x}(n)$ and $\mathbf{x}'(n)$ are different internal state vectors.

$$\begin{aligned}
&\left\|\mathbf{x}(n) - \mathbf{x}'(n)\right\| \\
&\quad = \left\|\mathbf{A}\mathbf{x}(n-1) + \mathbf{B}\mathbf{u}(n) - \mathbf{A}\mathbf{x}'(n-1) - \mathbf{B}\mathbf{u}(n)\right\| \\
&\quad = \left\|\mathbf{A}\mathbf{x}(n-1) - \mathbf{A}\mathbf{x}'(n-1)\right\| \\
&\quad \leq \|\mathbf{A}\| \left\|\mathbf{x}(n-1) - \mathbf{x}'(n-1)\right\| \\
&\quad = \left\|\mathbf{g}'\ \mathbf{W}\right\| \cdot \left\|\mathbf{x}(n-1) - \mathbf{x}'(n-1)\right\| \\
&\quad \leq \left\|\mathbf{g}'\right\| \cdot \|\mathbf{W}\| \cdot \left\|\mathbf{x}(n-1) - \mathbf{x}'(n-1)\right\| \\
&\quad \leq c \left\|\mathbf{x}(n-1) - \mathbf{x}'(n-1)\right\| \\
&\quad \leq c^2 \left\|\mathbf{x}(n-2) - \mathbf{x}'(n-2)\right\| \\
&\quad \leq \cdots \leq c^n \left\|\mathbf{x}(0) - \mathbf{x}'(0)\right\|.
\end{aligned}$$

This shows that the reservoir state depends on the external input and the effect of the initial state. The current reservoir state is determined by its past external input history, which guarantees the ESP.

V. SIMULATIONS AND RESULTS

In this section, the performance of the ALM-ESN is evaluated on the following chaotic time series: 1) the Lorenz chaotic time series prediction, 2) the Mackey-Glass chaotic time series prediction (MGS) and 3) the Sunspot series prediction. The normalized root mean square error (NRMSE) is used as the evaluation criteria of model performance [30]–[32], which is defined as follows:

$$\mathrm{NRMSE} = \sqrt{\sum_{t=1}^{S} \frac{(z_i(t) - y_i(t))^2}{S\sigma^2}}, \tag{24}$$

where $z_i(t)$ denotes the desired output, $y_i(t)$ is the network prediction output, σ^2 is the variance of the desired outputs, and S is the total number of $z_i(t)$.

To show the effectiveness of the ALM-ESN, the simulations are compared with the following models: OESN [8], SCR [33], DESN [34], RR-ESN [18] and Lasso-ESN [20]. The reservoir size, spectral radius, and sparsity are all obtained by the grid search method. All simulations are tested in MATLAB 2013b environment and run on i7-4790 with 3.60GHz CPU and 8.0GB RAM.

A. LORENZ CHAOTIC TIME SERIES PREDICTION

The Lorenz system can be described as follows [4]:

$$\begin{cases}\dot{x} = a_1(y - x), \\ \dot{y} = -xz + a_2 x - y, \\ \dot{z} = xy - a_3 z.\end{cases} \tag{25}$$

The typical system parameters can be chosen as $a_1 = 10$, $a_2 = 28$, $a_3 = 8/3$. In this case, the system is chaotic.

The fourth-order Runge-Kutta method is used to generate the data set. The initial values are selected as $x(0)$=1, $y(0)$=1, $z(0)$=0, and the step size is 0.01. To obtain the dynamic characteristic and predict $y(k+1)$, the embedded data vector $\alpha(k) = [y(k), y(k-8), y(k-2\times 8), \ldots, y(k-6\times 8)]^T$ is selected as in [4]. For the sample sequence pairs $\{\boldsymbol{\alpha}(k), y(k)\}$, the first 3000 values are used for training, the discarded values in training set are 1000, and the next 2000 values are used for testing.

After 100 independent simulations, the simulation results based on the ALM-ESN with different damping terms for the Lorenz chaotic time series are listed in TABLE 1. From TABLE 1, according to the training time and the testing NRMSE values, it can be found that it has better results when the damping term is chosen as $\mu_k = \alpha_k \left\|\mathbf{J}_k^T \mathbf{e}_k\right\|^\delta$, $\delta \in [1, 2]$. The testing outputs and errors comparing with different models for the Lorenz system are presented in Fig.4 and Fig.5, respectively. It can be obtained that the ALM-ESN has better accuracy and that the testing errors are limited in $[-4\times10^{-5}, 4\times10^{-5}]$. Based on the 100 independent simulations, the comparisons of the training time, the mean and variance of the testing NRMSE value with different approaches for the Lorenz system are described in TABLE 2. Obviously, the ALM-ESN has better performance than the other models according to the testing NRMSE values.

B. MACKEY-GLASS CHAOTIC TIME SERIES PREDICTION

The Mackey-Glass system (MGS) has been used as a standard benchmark model because of its chaotic characteristics,

TABLE 1. Simulation results based on ALM-ESN with different damping term for the Lorenz chaotic time series.

Damping term	Training	Testing NRMSE		Reservoir	Spectral	Sparsity
	time(s)	Mean	Variance	size	radius	
$\alpha_k \left\|\mathbf{J}_k^T \mathbf{e}_k\right\|^{\delta}$	**80.81**	**2.21×10^{-5}**	**3.21×10^{-6}**	200	0.8000	0.0250
$\alpha_k \left\|\mathbf{J}_k^T \mathbf{e}_k\right\|$	111.59	1.16×10^{-3}	1.76×10^{-4}	200	0.8000	0.0250
$\alpha_k \left\|\mathbf{e}_k\right\|^2$	113.73	1.89×10^{-3}	2.06×10^{-4}	200	0.8000	0.0250

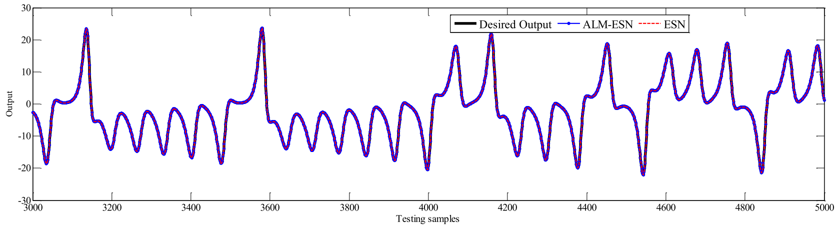

FIGURE 4. Testing outputs based on ALM-ESN and OESN for the Lorenz chaotic time series.

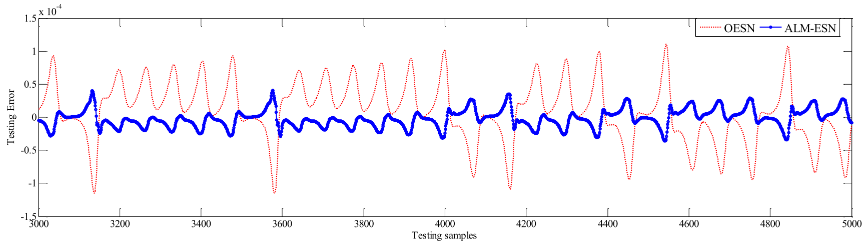

FIGURE 5. Testing error based on ALM-ESN and OESN for the Lorenz chaotic time series.

TABLE 2. Comparison of different models for the Lorenz chaotic time series.

Method	Training	Testing NRMSE		Reservoir size	Spectral	Sparsity
	time(s)	Mean	Variance		radius	
ALM-ESN	80.81	**2.21×10^{-5}**	**3.21×10^{-6}**	200	0.8000	0.0250
OESN[8]	79.42	9.19×10^{-4}	9.21×10^{-5}	400	0.8500	0.0450
SCR[33]	63.45	9.92×10^{-4}	8.31×10^{-5}	400	0.9500	0.0025
DESN[34]	81.68	7.72×10^{-4}	7.45×10^{-5}	400	0.9000	0.0238
RR-ESN[18]	82.32	8.38×10^{-4}	8.13×10^{-5}	400	0.9000	0.0300
Lasso-ESN[20]	-	5.19×10^{-4}	7.16×10^{-5}	400	0.8000	0.0300
BPSO-ESN[22]	93.56	5.23×10^{-4}	6.31×10^{-5}	400	0.8000	0.0300

on which the ESN has been successfully applied and shows good performance [8]. The MGS is derived from the following time-delay differential system [9]

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{ax(t-\tau)}{1+x^n(t-\tau)} + bx(t). \tag{26}$$

The MGS has a chaotic attractor when $\tau > 16.8$. The parameter values are selected as $n = 10, a = 0.2, b = -0.1, \tau = 17$ and the initial condition is $x(0)$=1.2 as in [9]. By the fourth-order Runge-Kutta method, 6000 samples are obtained. The number of the training samples is 3000, the first 1000 samples in training set are discarded to washout initial transient, and the number of testing samples is 2000.

The embedded data vector $\boldsymbol{\alpha}(k) = [x(k), x(k-6), x(k-2\times 6), \ldots, x(k-3\times 6)]^T$ is composed of four values of the time series as done in [9]. The target output is the 84-step ahead

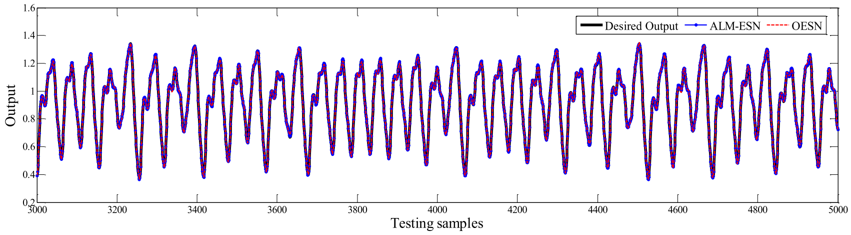

FIGURE 6. Testing outputs based on the ALM-ESN and OESN for the MGS.

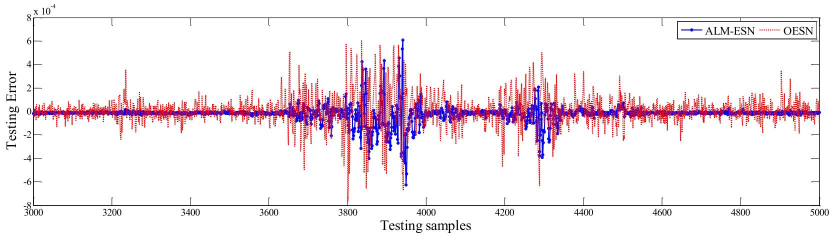

FIGURE 7. Testing error based on the ALM-ESN and OESN for the MGS.

TABLE 3. Comparison of different models for the MGS.

Method	Training time(s)	Testing NRMSE$_{84}$	Reservoir size	Spectral radius	Sparsity
ALM-ESN	151.21	**1.9268×10^{-4}**	200	0.8000	0.0200
OESN[8]	163.49	3.7925×10^{-4}	400	0.9000	0.0350
SCR[33]	**123.51**	4.6548×10^{-4}	300	0.8000	0.0033
DESN[34]	156.62	3.2968×10^{-4}	400	0.9000	0.0237
RR-ESN[18]	166.53	2.8813×10^{-4}	400	0.8500	0.0150
Lasso-ESN[20]	-	2.6532×10^{-4}	400	0.8500	0.0150
BPSO-ESN[22]	186.35	2.3823×10^{-4}	400	0.8500	0.0150

value of the time series. The network prediction performance is evaluated by the normalized root mean square error at the 84th time step (NRMSE$_{84}$) [9].

$$\mathrm{NRMSE}_{84} = \sqrt{\sum_{t=1}^{N_r} \frac{(z_i(84) - y_i(84))^2}{N_r \sigma^2}}. \tag{27}$$

where $z_i(84)$ denotes the 84-step target value, $y_i(84)$ is the corresponding network prediction value, σ^2 is the variance of the desired outputs and N_r is the number of independent simulations.

To validate the performance of the ALM-ESN, different methods are implemented. The testing outputs and error for the MGS are given in Fig.6 and Fig.7, respectively. It is known that the ALM-ESN fits very well and the testing errors are limited in $[-6\times10^{-4}, 6\times10^{-4}]$. Based on the 100 independent simulations of the MGS, the comparison of training time and testing NRMSE$_{84}$ and their relative parameters are listed in TABLE 3. Obviously, compared with other methods, the developed ALM-ESN has slightly better prediction performance than the other models according to the value of testing NRMSE$_{84}$.

C. SUNSPOT SERIES PREDICTION

The sunspot is one of the most basic and obvious solar activities on the Sun's photosphere, which can affect the earth's magnetic field. Therefore, it is significant to model and study sunspots [30]. The sunspot data are the monthly mean Wolf sunspot numbers in this simulation [35]. 3174 sets of data were collected from January 1749 to June 2013. The first 2200 values are used for training, the 200 discarded points are included in the training set, and the next 1174 values are used for testing. In this simulation, the embedded data vector is chosen as $\alpha(k) = [y(k), y(k-10), y(k-2\times10), \ldots, y(k-3\times10)]^T$ to predict the next value $y(k+1)$ as in [33], where $y(k)$ is the number of sunspots at time k.

The testing outputs and error for the sunspots are presented in Fig.8 and Fig.9, respectively. It can be obtained that the ALM-ESN has slightly better performance than the OESN. The detailed results are summarized in TABLE 4 based on

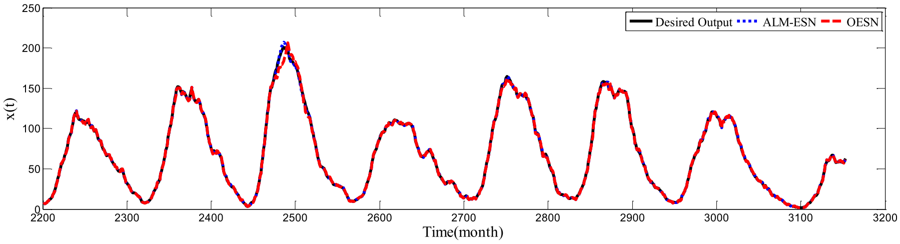

FIGURE 8. Testing outputs based on the ALM-ESN and the OESN for sunspots.

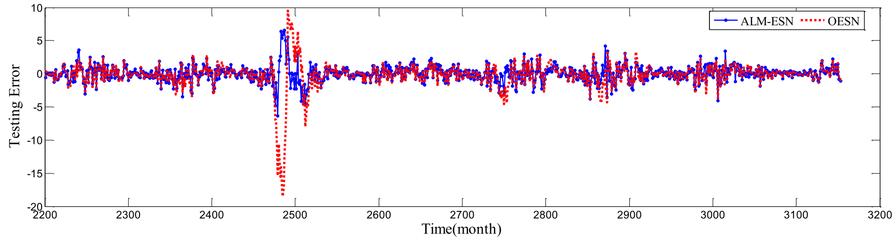

FIGURE 9. Testing error based on the ALM-ESN and the OESN for sunspots.

TABLE 4. Comparison of different models for sunspots.

Method	Training time(s)	Testing NRMSE		Reservoir size	Spectral radius	Sparsity
		Mean	Variance			
ALM-ESN	89.78	**0.1611**	0.0133	200	0.9500	0.0300
OESN[8]	90.33	0.2468	0.0424	300	0.9000	0.0500
SCR[33]	**79.03**	0.2604	0.0438	300	0.8500	0.0033
DESN[34]	85.12	0.2310	0.0325	300	0.9500	0.0238
RR-ESN[18]	95.23	0.2135	0.0289	300	0.8000	0.0450
Lasso-ESN[20]	-	0.1902	0.0232	300	0.8000	0.0450
BPSO-ESN[22]	109.23	0.1923	0.0276	300	0.8000	0.0450

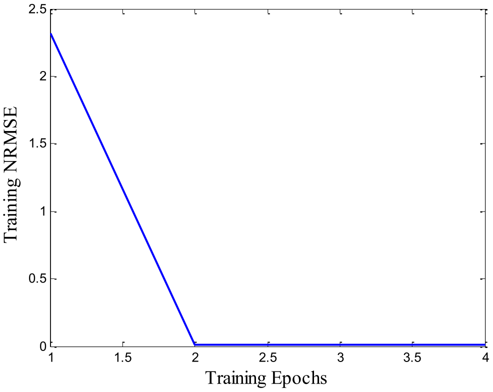

FIGURE 10. Training NRMSE curve for sunspots.

100 independent simulations. Based on the comparison of training time, the mean and variance of testing NRMSE in TABLE 4, the ALM-ESN has relatively higher prediction accuracy than the other models. The training NRMSE curve is shown in Fig.10, and the number of iterations is only 4.

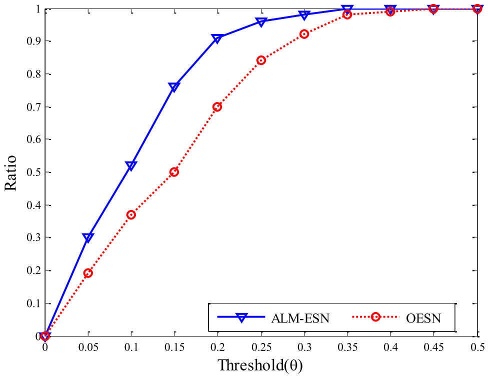

FIGURE 11. Successful design ratio based on the ALM-ESN and the OESN for sunspots.

To test the robustness of the ALM-ESN, the successful design ratio is introduced by

$$R(\theta) = \frac{\sum_{i=1}^{G} h(a_i - \theta)}{G}, \tag{28}$$

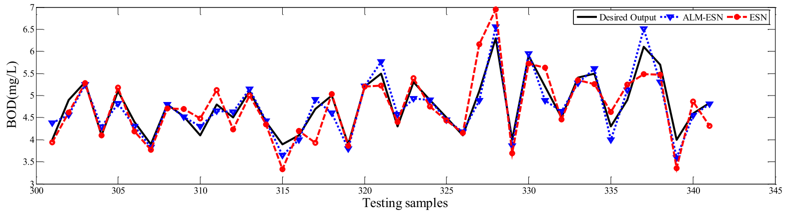

FIGURE 12. Testing outputs based on the ALM-ESN and the OESN for BOD.

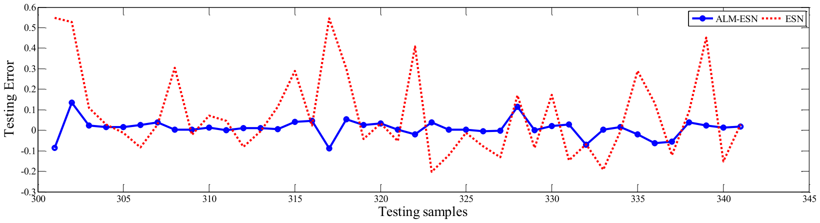

FIGURE 13. Testing error based on the ALM-ESN and the OESN for BOD.

TABLE 5. Comparison of different models for BOD.

Method	Training time(s)	Testing NRMSE		Reservoir size	Spectral radius	Sparsity
		Mean	Variance			
ALM-ESN	109.78	0.2710	**0.0255**	200	0.9500	0.0400
OESN[8]	70.33	0.3194	0.0652	300	0.8500	0.0350
SCR[33]	**60.93**	0.3602	0.0628	300	0.9500	0.0033
DESN[34]	75.12	0.3011	0.0525	300	0.9000	0.0238
RR-ESN[18]	78.23	0.2821	0.0492	300	0.8000	0.0250
Lasso-ESN[20]	-	0.2612	0.0312	300	0.8000	0.0250
BPSO-ESN[22]	123.68	**0.2435**	0.0289	300	0.8000	0.0250

$$h(x) = \begin{cases} 1 & x \leq 0 \\ 0 & x > 0 \end{cases} \tag{29}$$

where G is the number of experiments, and a_i is the prediction NRMSE for the ith experiment [34]. $R(\theta)$ is a probability estimation of obtaining a network whose prediction NRMSE is less than or equal to the threshold θ. For $R(\theta)$, the higher, the better.

After 100 independent experiments, the successful design ratios are presented in Fig.11. It is known that the ALM-ESN possesses higher successful design ratios than the OESN.

D. BIOCHEMICAL OXYGEN DEMAND PREDICTION IN THE WASTEWATER TREATMENT PROCESS

Wastewater treatment process (WWTP) is a complex system including a variety of physical and biochemical reactions. Due to the nonlinear characteristics, delay-time and uncertainty, it is difficult to measure effluent qualities parameters in the WWTP. Biochemical oxygen demand (BOD) is one of the most important effluent quality indexes and can reflect the water pollution situation. However, the conventional chemical measurement approaches cannot have a real time monitoring process. Therefore, water quality prediction model for BOD is essential to support water quality parameters. According to [36] and [37], Chemical Oxygen Demand (COD), suspended solids (SS), pH and dissolved oxygen (DO) are selected as the input variables. After deleting the abnormal data, 343 samples were got from a sewage treatment plant in Beijing, China. The first 200 values are used for training, and the next values are used for testing. The discarding points in training set are 50.

The testing results for effluent BOD are shown in Fig.12 and Fig.13, respectively, which illustrate that ALM-ESN has more accurate prediction than OESN for actual time-series. Based on 100 independent simulations, the detailed results are listed in Table 5. From the comparison of training time, mean values and variance of testing NRMSE in Table 5, ALM-ESN needs much training time, ALM-ESN still has high accuracy than OESN, SCR, DESN and RR-ESN, but has low accuracy than Lasso-ESN and BPSO-ESN.

VI. CONCLUSION

The ill-posed problem may occur in the learning process of the ESN. To solve this problem, an adaptive Levenberg-Marquardt algorithm based echo state network is developed. In the readout training process, the LM algorithm is used to replace the linear regression method for output weights, the damping term is selected adaptively, and the adaptive factor is amended by the trust region technique. Furthermore, to make the inputs fall within the active region of activation function, weight initialization is conducted to obtain the optimal region of initial weights using the Cauchy inequality. The simulation results for the three chaotic time-series predictions demonstrate that the ALM-ESN exhibits better prediction performance than some existing ESN construction methods.

REFERENCES

[1] C. Xiu, J. Hou, Y. Zang, G. Xu, and C. Liu, "Synchronous control of hysteretic creep chaotic neural network," *IEEE Access*, vol. 4, pp. 8617–8624, Dec. 2016.

[2] X. Bai, F. Zhang, J. Hou, F. Xia, A. Tolba, and E. Elashkar, "Implicit multi-feature learning for dynamic time series prediction of the impact of institutions," *IEEE Access*, vol. 5, pp. 16372–16382, Aug. 2017.

[3] H. Han, X.-L. Wu, and J.-F. Qiao, "Nonlinear systems modeling based on self-organizing fuzzy-neural-network with adaptive computation algorithm," *IEEE Trans. Cybern.*, vol. 44, no. 4, pp. 554–564, Apr. 2014.

[4] D. Li, M. Han, and J. Wang, "Chaotic time series prediction based on a novel robust echo state network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 787–799, May 2012.

[5] H.-G. Han, Z.-L. Lin, and J.-F. Qiao, "Modeling of nonlinear systems using the self-organizing fuzzy neural network with adaptive gradient algorithm," *Neurocomputing*, vol. 266, pp. 566–578, Nov. 2017.

[6] 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.

[7] R. Chandra, "Competition and collaboration in cooperative coevolution of elman recurrent neural networks for time-series prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3123–3136, Dec. 2015.

[8] 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.

[9] J. Qiao, F. Li, H. Han, and W. Li, "Growing echo-state network with multiple subreservoirs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 391–404, Feb. 2017.

[10] Z. K. Malik, A. Hussain, and Q. J. Wu, "Multilayered echo state machine: A novel architecture and algorithm," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 946–959, Apr. 2017.

[11] M. Chen, W. Saad, and C. Yin, "Echo state networks for self-organizing resource allocation in LTE-U with uplink–downlink decoupling," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 3–16, Jan. 2017.

[12] J. Park, B. Lee, S. Kang, P. Y. Kim, and H. J. Kim, "Online learning control of hydraulic excavators based on echo-state networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 249–259, Jan. 2017.

[13] F. M. Bianchi, E. De Santis, A. Rizzi, and A. Sadeghian, "Short-term electric load forecasting using echo state networks and PCA decomposition," *IEEE Access*, vol. 3, pp. 1931–1943, 2015.

[14] H. Wang, C. Ni, and X. Yan, "Optimizing the echo state network based on mutual information for modeling fed-batch bioprocesses," *Neurocomputing*, vol. 225, pp. 111–118, Feb. 2017.

[15] Y.-F. Yam, C.-T. Leung, P. K. S. Tam, and W.-C. Siu, "An independent component analysis based weight initialization method for multilayer perceptrons," *Neurocomputing*, vol. 48, nos. 1–4, pp. 807–818, Oct. 2002.

[16] J. Qiao, S. Li, and W. Li, "Mutual information based weight initialization method for sigmoidal feedforward neural networks," *Neurocomputing*, vol. 207, pp. 676–683, Sep. 2016.

[17] K. Xiang, B. N. Li, L. Zhang, M. Pang, M. Wang, and X. Li, "Regularized Taylor echo state networks for predictive control of partially observed systems," *IEEE Access*, vol. 4, pp. 3300–3309, Jun. 2016.

[18] X. Dutoit, B. Schrauwen, J. Van Campenhout, D. Stroobandt, H. Van Brussel, and M. Nuttin, "Pruning and regularization in reservoir computing," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1534–1546, Mar. 2009.

[19] R. F. Reinhart and J. J. Steil, "Regularization and stability in reservoir networks with output feedback," *Neurocomputing*, vol. 90, no. 8, pp. 96–105, Aug. 2012.

[20] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[21] S. Zhong, X. Xie, L. Lin, and F. Wang, "Genetic algorithm optimized double-reservoir echo state network for multi-regime time series prediction," *Neurocomputing*, vol. 238, pp. 191–204, May 2017.

[22] H. Wang and X. Yan, "Optimizing the echo state network with a binary particle swarm optimization algorithm," *Knowl.-Based Syst.*, vol. 86, pp. 182–193, Sep. 2015.

[23] M. Xu and M. Han, "Adaptive elastic echo state network for multivariate time series prediction," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2173–2183, Jul. 2016.

[24] J.-M. Wu, "Multilayer potts perceptrons with Levenberg–Marquardt learning," *IEEE Trans. Neural Netw.*, vol. 19, no. 12, pp. 2032–2043, Dec. 2008.

[25] T. Xie, H. Yu, J. Hewlett, P. Rozycki, and B. Wilamowski, "Fast and efficient second-order method for training radial basis function networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 609–619, Apr. 2012.

[26] N. Yamashita and M. Fukushima, "On the rate of convergence of the Levenberg–Marquardt method," *Computing*, vol. 15, pp. 239–249, May 2001.

[27] C. Ma and J. Jiang, "Some research on Levenberg–Marquardt method for the nonlinear equations," *Appl. Math. Comput.*, vol. 184, no. 2, pp. 1032–1040, Jan. 2007.

[28] M. J. D. Powell, "Convergence properties of a class of minimization algorithms," *Nonlinear Programm.*, vol. 2, pp. 1–27, Apr. 1975.

[29] J. Fan and J. Zeng, "A Levenberg–Marquardt algorithm with correction for singular system of nonlinear equations," *Appl. Math. Comput.*, vol. 219, no. 17, pp. 9438–9446, May 2013.

[30] H. Wang and X. Yan, "Improved simple deterministically constructed cycle reservoir network with sensitive iterative pruning algorithm," *Neurocomputing*, vol. 145, pp. 353–362, Dec. 2014.

[31] K. Gu, G. Zhai, W. Lin, X. Yang, and W. Zhang, "No-reference image sharpness assessment in autoregressive parameter space," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3218–3231, Oct. 2015.

[32] K. Gu, G. Zhai, X. Yang, and W. Zhang, "Hybrid no-reference quality metric for singly and multiply distorted images," *IEEE Trans. Broadcast.*, vol. 60, no. 3, pp. 555–567, Sep. 2014.

[33] A. Rodan and P. Tino, "Minimum complexity echo state network," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 131–144, Jan. 2011.

[34] Y. Xue, L. Yang, and S. Haykin, "Decoupled echo state networks with lateral inhibition," *Neural Netw.*, vol. 20, no. 3, pp. 365–376, Apr. 2007.

[35] (2014). *National Geophysical Data Center Sunspot Numbers*. [Online]. Available: http://www.ngdc.noaa.gov/stp/space-weather/solar-data/solarsolarindices/sunspot-numbers/international/tables/

[36] F. Li, J. Qiao, H. Han, and C. Yang, "A self-organizing cascade neural network with random weights for nonlinear system modeling," *Appl. Soft Comput.*, vol. 42, pp. 184–193, May 2016.

[37] J. Qiao, Z. Hu, and W. Li, "Soft measurement modeling based on chaos theory for biochemical oxygen demand (BOD)," *Water*, vol. 8, no. 12, p. 581, Dec. 2016.

JUNFEI QIAO received the B.E. and M.E. degrees in control engineering from Liaoning Technical University, Fuxin, China, in 1992 and 1995, respectively, and the Ph.D. degree from Northeast University, Shenyang, China, in 1998. From 1998 to 2000, he was a Post-Doctoral Fellow with the School of Automatics, Tianjin University, Tianjin, China. He is currently a Professor with the Beijing University of Technology, Beijing, China. His current research interests include neural networks, intelligent systems, self-adaptive/learning systems, and process control systems.

LEI WANG received the B.E. degree in mathematics from Qufu Normal University, Qufu, China, in 2003, and the M.E. degree in mathematics from the Beijing University of Technology, Beijing, China, in 2006. He is currently pursuing the Ph.D. degree with the Beijing University of Technology. His current research interests include neural networks and self-adaptive learning systems.

CUILI YANG received the M.S. degree in control theory and control engineering from Tianjin University, Tianjin, China, in 2010, and the Ph.D. degree in control theory and control engineering from the City University of Hong Kong, Hong Kong, in 2014. She is currently a Lecturer with the Beijing University of Technology. Her current research interests include computational intelligence, modeling, and control for wastewater treatment process.

KE GU received the B.S. and Ph.D. degrees in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009 and 2015, respectively. He has published 15 IEEE Transactions papers as the first author and totally published nearly 30 IEEE Transactions papers. His research interests include quality assessment, environmental perception, image processing, machine learning, and big data. He received the Best Paper Award at the IEEE International Conference on Multimedia and Expo in 2016, and received the excellent Ph.D. thesis award from the Chinese Institute of Electronics, in 2016. He is the leading special session organizers in VCIP2016 and ICIP2017. He also serves as the long-term reviewer for over 20 top SCI Journals. He is currently an Associate Editor for the IEEE Access.