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. 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 aresmall, 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 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 algorithms 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.
be rewritten as

\[ W^{\text{out}} = (X^+Z)^T = (BA^{-1}A^TZ)^T = \sum_{i=1}^{Q} \frac{1}{\lambda_i} b(i) a(i)^T z(i) \]  

(4)

where \( a(i) \) and \( b(i) \) are the \( i \)-th column of \( A \) and \( B \), respectively. From formula (4), it is known that if the smallest singular value of \( X \) 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 \( \frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^d(t-\tau)} + bx(t) \) 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 \( \beta \) increases, the mean testing error fluctuates. The mean testing error is at its minimum when \( \beta \approx 0.3 \), 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.](image)

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

\[
\begin{align*}
\sigma &\leq \sum_{j=1}^{N} w_{ik,j,p-1} + \sum_{k=1}^{K} w_{ik}^m u_{kp} \leq \bar{s} \\
(5)
\end{align*}
\]

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

\[
x_{j,p-1}(p = 1, \cdots, P) \text{ is the } j \text{-th element of } x(p-1) (p = 1, \cdots, P).
\]

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

\[
\begin{align*}
\left( \sum_{k=1}^{K} (u_{kp})^2 + \sum_{j=1}^{N} (x_{j,p-1})^2 \right) \left( \sum_{k=1}^{K} (w_{ik}^m)^2 + \sum_{j=1}^{N} (w_{ij})^2 \right) \leq \bar{s}^2,
\end{align*}
\]

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

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

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

\[
\begin{align*}
\left( \sum_{j=1}^{N} w_{ik} x_{j,p-1} + \sum_{k=1}^{K} w_{ik}^m u_{kp} \right)^2 \leq \bar{s}^2.
\end{align*}
\]

Using the Cauchy inequality,

\[
\begin{align*}
\left( \sum_{j=1}^{N} w_{ik} x_{j,p-1} + \sum_{k=1}^{K} w_{ik}^m u_{kp} \right)^2 \leq \left( \sum_{k=1}^{K} (u_{kp})^2 \right) \left( \sum_{j=1}^{N} (w_{ik})^2 \right) \leq \bar{s}^2.
\end{align*}
\]  

(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

\[
\begin{align*}
\sum_{k=1}^{K} (w_{ik})^2 \approx E \left( \sum_{k=1}^{K} (w_{ik})^2 \right) = K \cdot \text{var}(w_{ik}) = K \frac{\beta_p^2}{3}.
\end{align*}
\]

(7)
Similarly,
\[ \sum_{j=1}^{N} (w_{ij})^2 \approx \frac{N \beta_p^2}{3}. \]  
(8)

Then
\[ \beta_p \leq \delta \left( \frac{3}{(K + N) \left( \sum_{k=1}^{K} (n_k p)^2 + \sum_{j=1}^{N} (x_j p - 1)^2 \right)} \right), \]
(9)

Let
\[ \beta = \min_{p=1, \ldots, P} \beta_p. \]
(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:
\[ E(\mathbf{W}^{out}) = \frac{1}{2} \sum_{p=1}^{P} \sum_{j=1}^{M} (y_j^p - \mathbf{y}_j)^2 = \frac{1}{2} \sum_{q=1}^{Q} e_q^2, \]
(11)

where \( e_q = y_j^q - \mathbf{y}_j \), \( y_j \) is the desired output, and \( \mathbf{y}_j \) 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:
\[ \mathbf{W}^{out}(k + 1) = \mathbf{W}^{out}(k) - (\mathbf{J}_k^T \mathbf{J}_k + \mu_k \mathbf{I})^{-1} \mathbf{J}_k^T e_k, \]
(12)

where \( \mathbf{J}_k \) is a Jacobi matrix, \( e_k \) is an error vector, and \( \mu_k \) is a positive damping term.

\[ e_k \triangleq e(\mathbf{W}^{out}(k)) = (e_1, e_2, \ldots, e_Q)^T, \]
(13)

\[ \mathbf{J}_k \triangleq \mathbf{J}(\mathbf{W}^{out}(k)) = \begin{pmatrix} \frac{\partial e_1}{\partial w_{11}^{out}} & \frac{\partial e_1}{\partial w_{12}^{out}} & \cdots & \frac{\partial e_1}{\partial w_{1Q}^{out}} \\ \frac{\partial e_2}{\partial w_{11}^{out}} & \frac{\partial e_2}{\partial w_{12}^{out}} & \cdots & \frac{\partial e_2}{\partial w_{1Q}^{out}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_Q}{\partial w_{11}^{out}} & \frac{\partial e_Q}{\partial w_{12}^{out}} & \cdots & \frac{\partial e_Q}{\partial w_{1Q}^{out}} \\ \frac{\partial e_1}{\partial w_{K+1}^{out}} & \frac{\partial e_2}{\partial w_{K+1}^{out}} & \cdots & \frac{\partial e_Q}{\partial w_{K+1}^{out}} \\ \frac{\partial e_1}{\partial w_{K+1}^{out}} & \frac{\partial e_2}{\partial w_{K+1}^{out}} & \cdots & \frac{\partial e_Q}{\partial w_{K+1}^{out}} \end{pmatrix}. \]
(14)

According to (12), since \( \mu_k \) 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 \( \mu_k \). However, there is no general rule in the selecting method of the damping term. In [26], the parameter is chosen as \( \mu_k = \|e_k\|^2 \), and it can be shown that the LM algorithm possesses quadratic convergence. However, the damping term \( \mu_k = \|e_k\|^2 \) (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, \( \mu_k = \|e_k\|^2 \) may be less than the machine accuracy. Therefore, it may have no effect. Moreover, \( \mu_k = \|e_k\|^2 \) 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 \( \mu_k = \theta \|e_k\| + (1 - \theta) \|\mathbf{J}_k^T e_k\| \theta \in (0, 1) \) 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 \( \mu_k = \alpha_k \|\mathbf{J}_k^T e_k\|^\delta \) with \( \delta \in [1, 2] \), where \( \alpha_k \) 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:
\[ \text{Actual} \mathbf{d}_k = \|e_k\|^2 - \|e_k + \mathbf{J}_k \mathbf{d}_k\|^2, \]
(15)
\[ \text{Predict} \mathbf{d}_k = \|e_k\|^2 - \|e_k + \mathbf{J}_k \mathbf{d}_k\|^2. \]
(16)

The ratio \( r_k = \frac{\text{Actual} \mathbf{d}_k}{\text{Predict} \mathbf{d}_k} \) is important to adopt the trial step and update the parameter \( \alpha_k \) between these reductions.

Based on the LM algorithm, each iteration may be described 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 \( \mathbf{W} = (\alpha_{W0}/\rho(\mathbf{W}_0))\mathbf{W}_0 \), where \( 0 < \alpha_{W} < 1 \) and \( \rho(\mathbf{W}_0) \) is the spectral radius of \( \mathbf{W}_0 \). Initialize the internal state \( x(0) \).

**Step 3:** Randomly produce an input weight matrix \( \mathbf{W}_n \) 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 \( n_{\text{min}} \).

**Step 5:** Compute the network output, the error vector \( e_k \), 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 \( \|\mathbf{J}_k^T e_k\| \leq \varepsilon \), stop; otherwise compute \( \mathbf{d}_k = -\alpha_k (\mathbf{J}_k^T \mathbf{J}_k + \mu_k \mathbf{I})^{-1} \mathbf{J}_k^T e_k \), where \( \mu_k = \alpha_k \|\mathbf{J}_k^T e_k\|^\delta, \delta \in [1, 2], \varepsilon = 10^{-6}, \alpha_1 = 10^{-7}, m = 10^{-8}, p_0 = 0.0001, p_1 = 0.25, p_2 = 0.75 \).
Step 7: Compute \( r_k = \text{Ared}_k / \text{Pred}_k \), let
\[
\text{W}_{\text{out}}(k + 1) = \begin{cases} 
\text{W}_{\text{out}}(k) + \text{d}_k, & \text{if } r_k > p_0, \\
\text{W}_{\text{out}}(k), & \text{otherwise}.
\end{cases}
\]

Step 8: Compute
\[\alpha_{k+1} = \begin{cases} 
4\alpha_k, & \text{if } r_k < p_1, \\
\alpha_k, & \text{if } r_k \in [p_1, p_2], \\
\max(\frac{\alpha_k}{4}, m), & \text{if } r_k > p_2.
\end{cases}\]
go to Step 5.

Step 9: Test the trained ALM-ESN.

### III. CONVERGENCE ANALYSIS

Let
\[ e(\text{W}_{\text{out}}) = 0. \]  

Suppose the solution of (18) is nonempty and denote by \( \Omega \).
It is obvious that
\[ \text{d}_k = -(J_k^T J_k + \mu_k I)^{-1} J_k^T \text{e}_k. \]  

is a solution of
\[
\min_{\text{d}} \|J_k \text{d} + \text{e}_k\|^2 + \mu_k \|\text{d}\|^2.
\]

Define
\[ \Delta_k = \|J_k^T J_k + \mu_k I\|^{-1} J_k^T \text{e}_k. \]  

It can be determined that (12) is equivalent to the following trust region problem:
\[
\min_{\text{d}} \|J_k \text{d} + \text{e}_k\|^2 \\
\text{s.t. } \|\text{d}\| \leq \Delta_k.
\]

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:** \( \text{e}_k \) is continuously differentiable. Both \( \text{e}_k \) and its Jacobian matrix \( J_k \) are Lipschitz continuous, i.e., there exist positive constants \( L_1 \) and \( L_2 \) such that
\[
\begin{align*}
&\forall \text{W}_{\text{out}}^1, \text{W}_{\text{out}}^2 \in \text{W}_{\text{out}}, \\
&\forall \text{W}_{\text{out}}^1, \text{W}_{\text{out}}^2 \in \text{W}_{\text{out}}.
\end{align*}
\]

By Assumption 3, it can be obtained
\[
\begin{align*}
&\|J(\text{W}_{\text{out}}^1) - J(\text{W}_{\text{out}}^2)\| \leq L_1 \|\text{W}_{\text{out}}^1 - \text{W}_{\text{out}}^2\|, \\
&\forall \text{W}_{\text{out}}^1, \text{W}_{\text{out}}^2 \in \text{W}_{\text{out}}.
\end{align*}
\]

**Assumption 4:** \( \|e(\text{W}_{\text{out}})\| \) provides a local error bound on \( N(\text{W}_{\text{out}}, b_1) \) for (17), i.e., there exist two constants \( c_1 > 0 \) and \( b_1 < 1 \) such that
\[ \|e(\text{W}_{\text{out}})\| \geq c_1 \text{dist}(\text{W}_{\text{out}}, \Omega), \quad \forall \text{W}_{\text{out}} \in \text{W}_{\text{out}}. \]

where \( \text{dist}(\text{W}_{\text{out}}, \Omega) = \min_{\text{W}_{\text{out}} \in \Omega} \|\text{W}_{\text{out}} - \text{W}_{\text{out}}\|. \)

\[ N(\text{W}_{\text{out}}, b_1) = \{ \text{W}_{\text{out}} \mid \|\text{W}_{\text{out}} - \text{W}_{\text{out}}\| \leq b_1 \}, \text{W}_{\text{out}} \in \text{W}_{\text{out}}. \]

**Lemma 5 [28]:** Let \( \text{d}_k \) be computed by Algorithm 2. Then the predicted reduction satisfies
\[ \text{Pred}_k \geq \|J_k^T \text{e}_k\| \min \{\|\text{d}_k\|, \|J_k^T \text{e}_k\| / \|J_k^T J_k\|\}. \]

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

**Theorem 6:** The sequence \( \{\text{W}_{\text{out}}(k)\} \) generated by Algorithm 2 satisfies \( \lim_{k \to \infty} \|J_k^T \text{e}_k\| = 0. \)

**Proof:** If the theorem is not true, then there exists a constant \( \tau > 0 \) and infinitely many \( k \) such that \( \|J_k^T \text{e}_k\| \geq \tau \).

Let
\[ \begin{align*}
&K = \{k \mid \|J_k^T \text{e}_k\| \geq \tau\}, \\
&T = \{k \mid \text{W}_{\text{out}}(k + 1) \neq \text{W}_{\text{out}}(k), k \in K\}.
\end{align*} \]

Using Lemma 5, it can be got that
\[ \|e_k\|^2 \geq \sum_{k \in K} (\|e_k\|^2 - \|e_{k+1}\|^2) = \sum_{k \in T} (\|e_k\|^2 - \|e_{k+1}\|^2) \geq \sum_{k \in T} p_0 \text{Pred}_k \geq \sum_{k \in T} p_0 \|J_k^T \text{e}_k\| \times \min\{\|\text{d}_k\|, \|J_k^T \text{e}_k\| / \|J_k^T J_k\|\}. \]

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

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

Additionally, it follows from \( \|J_k \text{e}_k\| \geq \tau, \sum_{k \geq \hat{k}} \|\text{d}_k\| < +\infty \) for all \( k \geq \hat{k} \) and Lemma 5 that
\[ r_k = \frac{\text{Ared}_k}{\text{Pred}_k} = 1 + \frac{\|e_k + J_k \text{d}_k\| O(\|\text{d}_k\|^2)}{\text{Pred}_k} \leq 1 + \|e_k + J_k \text{d}_k\| O(\|\text{d}_k\|^2) \leq 1 + \frac{O(\|\text{d}_k\|^2)}{\|\text{d}_k\|} \to 1. \]

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 \( \text{W}_{\text{out}}(k) \) is sufficiently close to \( \Omega \), i.e., \( \text{dist}(\text{W}_{\text{out}}(k), \Omega) \ll 1 \). Let \( \text{W}_{\text{out}}(k) \in \Omega \) satisfy \( \|\text{W}_{\text{out}}(k) - \hat{\text{W}}_{\text{out}}(k)\| = \text{dist}(\text{W}_{\text{out}}, \Omega) \).

**Lemma 7 [27]:** If \( \text{W}_{\text{out}}(k) \in N(\text{W}_{\text{out}}, b_1) \), then there is a constant \( c_2 > 0 \) satisfying
\[ c_2 \text{dist}(\text{W}_{\text{out}}, \Omega)^\delta \leq \mu_k \]
\[ = \alpha_k \|J_k^T \text{e}_k\| \leq L_2 \|\text{W}_{\text{out}}(k) - \text{W}_{\text{out}}(k)\|^\delta. \]

**Lemma 8 [27]:** \( \|\text{d}_k\| \leq O(\|\text{W}_{\text{out}}(k) - \text{W}_{\text{out}}(k)\|). \)
Using the SVD of the Jacobi matrix, the quadratic convergence of Algorithm 2 is studied. Suppose that the SVD of $J(W_{\text{out}})$ is

$$J(W_{\text{out}}) = U^* \Sigma^* V^* = (U^*_1, U^*_2) \left( \Sigma^*_1 \begin{pmatrix} \Sigma^*_2 \end{pmatrix} \begin{pmatrix} O \end{pmatrix} \right) \begin{pmatrix} V^T_1 \end{pmatrix}$$

where $\Sigma^*_1 = \text{diag}(\sigma^*_1, \ldots, \sigma^*_r), \sigma^*_1 \geq \sigma^*_2 \geq \ldots \geq \sigma^*_r > 0, \text{rank}((\Sigma^*_1) \geq r)$. Suppose that the SVD of $J(W_{\text{out}}(k)) = J_k$ is as follows.

$$J_k = U_k \Sigma_k V_k^T$$

$$= (U_{k,1}, U_{k,2}, U_{k,3}) \begin{pmatrix} \Sigma_{k,1} & \Sigma_{k,2} & O \end{pmatrix} \begin{pmatrix} V^T_{k,1} & V^T_{k,2} & V^T_{k,3} \end{pmatrix}$$

where $\Sigma_{k,1} = \text{diag}(\sigma^{(k)}_1, \ldots, \sigma^{(k)}_r), \Sigma_{k,2} = \text{diag}(\sigma^{(k)}_r, \ldots, \sigma^{(k)}_{r+q}), \sigma^{(k)}_1 \geq \sigma^{(k)}_2 \geq \ldots \sigma^{(k)}_r \geq \sigma^{(k)}_{r+q} > 0, q \geq 0$.

For convenience, denote $\Sigma_{k,i}, U_{k,i}$ and $V_{k,i}$ as $\Sigma_i, U_i$ and $V_i$ $(i = 1, 2, 3)$, respectively. Consequently, the SVD of $J_k$ can be written as $J_k = U_1 \Sigma_1 V_1^T + U_2 \Sigma_2 V_2^T$.

Lemma 9 [29]:

(a) $\|U_1 U_1^T e_k\| \leq O(\|W_{\text{out}}(k) - \tilde{W}_{\text{out}}(k)\|)$;

(b) $\|U_2 U_2^T e_k\| \leq O(\|W_{\text{out}}(k) - \tilde{W}_{\text{out}}(k)\|)$;

(c) $\|U_3 U_3^T e_k\| \leq O(\|W_{\text{out}}(k) - \tilde{W}_{\text{out}}(k)\|^2)$.

The local convergence of Algorithm 2 is given as follows.

Theorem 10: The sequence $\{W_{\text{out}}(k)\}$ 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 $\|W_{\text{out}}(k) - \tilde{W}_{\text{out}}(k)\| \leq d_k$, from Lemma 8, it can be got that

$$\|W_{\text{out}}(k) - W_{\text{out}}(k)\| = O(\|d_k\|),$$

$$\|e_k\| = \|e_k + J_k d_k\| \geq \|e_k\| - \|e_k + J_k (\tilde{W}_{\text{out}}(k) - W_{\text{out}}(k))\| \geq c_1 \|\tilde{W}_{\text{out}}(k) - W_{\text{out}}(k)\| = O(\|d_k\|).$$

2) If $\|W_{\text{out}}(k) - \tilde{W}_{\text{out}}(k)\| > d_k$,

$$\|e_k\| = \|e_k + J_k d_k\| \geq \|e_k\| - \|e_k + J_k (\tilde{W}_{\text{out}}(k) - W_{\text{out}}(k))\| \geq \|W_{\text{out}}(k) - \tilde{W}_{\text{out}}(k)\|$$

$$\times (\|e_k\| - \|e_k + J_k (\tilde{W}_{\text{out}}(k) - W_{\text{out}}(k))\|) \geq \|W_{\text{out}}(k) - \tilde{W}_{\text{out}}(k)\|$$

$$\times (\|e_k\| - \|e_k + J_k (\tilde{W}_{\text{out}}(k) - W_{\text{out}}(k))\|) \geq \|W_{\text{out}}(k) - \tilde{W}_{\text{out}}(k)\| (c_1 \|\tilde{W}_{\text{out}}(k) - W_{\text{out}}(k)\|^2 + O(\|\tilde{W}_{\text{out}}(k) - W_{\text{out}}(k)\|^2))$$

$$\geq c_1 \|d_k\|,$$

Since $\{W_{\text{out}}(k)\}$ converges to $W_{\text{out}}$, assume that $L_1 \|W_{\text{out}}(k) - W_{\text{out}}\| \leq \frac{\sigma_f}{2}$.

$$\left(\sum^2 + \mu_k I\right)^{-1} \leq \sum^{-2} \leq \frac{4}{\sigma_f^2} \leq \mu_k^{-1}.$$

Since $\{W_{\text{out}}(k)\}$ converges to $W_{\text{out}}$, assume that $L_1 \|W_{\text{out}}(k) - W_{\text{out}}\| \leq \frac{\sigma_f}{2}$.

$$\|e_k + J_k d_k\| \leq O(\|\tilde{W}_{\text{out}}(k) - W_{\text{out}}(k)\|^2),$$

$$\|e_k + J_k d_k\| \leq O(\|W_{\text{out}}(k) - W_{\text{out}}(k)\|^2).$$

$c_1 \text{dist}(W_{\text{out}}(k + 1), \Omega) \leq \|e_{k+1}\| = \|e_{k+1} + e_k + J_k d_k - e_k - J_k d_k\| \leq \|e_k + J_k d_k\| + \|e_{k+1} - e_k - J_k d_k\| \leq \|e_k + J_k d_k\| + O(\|d_k\|^2) \leq O(\|W_{\text{out}}(k) - W_{\text{out}}(k)\|^2) \leq O(\|W_{\text{out}}(k) - W_{\text{out}}(k)\|^2).$
It follows from \( \| W^{out}(k) - W^{out}(k) \| \leq \| d_k \| + \| W^{out}(k + 1) - W^{out}(k + 1) \| \) that \( \| W^{out}(k) - W^{out}(k) \| \leq 2 \| d_k \| \) holds for all sufficiently large \( k \).

\[ \| d_{k+1} \| = O(\| d_k \|^2), \]

which implies that \( \{ W^{out}(k) \} \) converges quadratically to \( W_\ast \), namely

\[ \| W^{out}(k + 1) - W_\ast \| = O(\| W^{out}(k) - W_\ast \|^2). \]

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 reservoir matrix \( \chi \). The sufficient condition is that the maximal singular value of \( \chi \) is less than 1 (\( \sigma(\chi) < 1 \)). Since \( \| \chi \| = \sigma(\chi) \), the sufficient condition is equivalent to \( c \triangleq \| \chi \| < 1 \).

Suppose that \( x(n) \) and \( x'(n) \) are different internal state vectors.

\[
\begin{align*}
\| x(n) - x'(n) \| &= \| A x(n) - A x'(n) + B u(n) - B u(n) \| \\
&= \| A x(n) - A x'(n) \| \\
&\leq \| A \| \| x(n) - x'(n) \| \\
&= \| g \chi \| \| x(n) - x'(n) \| \\
&\leq \| g \| \| \chi \| \| x(n) - x'(n) \| \\
&\leq \| g \| \| \chi \| \| x(n) - x'(n) \| \\
&\leq \| g \| \| \chi \| \| x(n) - x'(n) \| \\
&\leq \cdot \cdot \cdot \leq \| x(0) - x'(0) \|. \\
\end{align*}
\]

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:

\[
\text{NRMSE} = \frac{\sum_{i=1}^{s} (z_i(t) - y_i(t))^2}{\sigma^2 S}.
\]

where \( z_i(t) \) denotes the desired output, \( y_i(t) \) is the network prediction output, \( \sigma^2 \) 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]:

\[
\begin{align*}
\dot{x} &= a(y - x), \\
\dot{y} &= -xz + ay, \\
\dot{z} &= xy - az, \\
\end{align*}
\]

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 \( \{\alpha(k), y(k)\} \), 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 \( \mu_k = \alpha_k \| F_k e_k \|^2, \delta \in [1, 2] \). 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 \([-4 \times 10^{-5}, 4 \times 10^{-5}]\). 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.

<table>
<thead>
<tr>
<th>Damping term</th>
<th>Training time(s)</th>
<th>Testing NRMSE</th>
<th>Reservoir size</th>
<th>Spectral radius</th>
<th>Sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
<td>Variance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha_k | \mathbf{e}_t |^2$</td>
<td>80.81</td>
<td>$2.21 \times 10^5$</td>
<td>$3.21 \times 10^4$</td>
<td>200</td>
<td>0.8000</td>
</tr>
<tr>
<td>$\alpha_k | \mathbf{e}_t |^2$</td>
<td>111.59</td>
<td>$1.16 \times 10^4$</td>
<td>$1.76 \times 10^4$</td>
<td>200</td>
<td>0.8000</td>
</tr>
<tr>
<td>$\alpha_k | \mathbf{e}_t |^2$</td>
<td>113.73</td>
<td>$1.89 \times 10^3$</td>
<td>$2.06 \times 10^4$</td>
<td>200</td>
<td>0.8000</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Training time(s)</th>
<th>Testing NRMSE</th>
<th>Reservoir size</th>
<th>Spectral radius</th>
<th>Sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
<td>Variance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALM-ESN</td>
<td>80.81</td>
<td>$2.21 \times 10^5$</td>
<td>$3.21 \times 10^4$</td>
<td>200</td>
<td>0.8000</td>
</tr>
<tr>
<td>OESN[9]</td>
<td>79.42</td>
<td>$9.19 \times 10^4$</td>
<td>$9.21 \times 10^5$</td>
<td>400</td>
<td>0.8500</td>
</tr>
<tr>
<td>SCR[9]</td>
<td>63.45</td>
<td>$9.92 \times 10^4$</td>
<td>$8.31 \times 10^4$</td>
<td>400</td>
<td>0.9500</td>
</tr>
<tr>
<td>DESN[9]</td>
<td>81.68</td>
<td>$7.72 \times 10^4$</td>
<td>$7.45 \times 10^5$</td>
<td>400</td>
<td>0.9000</td>
</tr>
<tr>
<td>RR-ESN[9]</td>
<td>82.32</td>
<td>$8.38 \times 10^4$</td>
<td>$8.13 \times 10^5$</td>
<td>400</td>
<td>0.9000</td>
</tr>
<tr>
<td>Lasso-ESN[20]</td>
<td>-</td>
<td>$5.19 \times 10^4$</td>
<td>$7.16 \times 10^4$</td>
<td>400</td>
<td>0.8000</td>
</tr>
<tr>
<td>BPSO-ESN[22]</td>
<td>93.56</td>
<td>$5.23 \times 10^4$</td>
<td>$6.31 \times 10^5$</td>
<td>400</td>
<td>0.8000</td>
</tr>
</tbody>
</table>

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]

$$\frac{dx(t)}{dt} = \frac{ax(t - \tau)}{1 + x^2(t - \tau)} + bx(t).$$  \hspace{1cm} (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 $\mathbf{e}(k) = [x(k), x(k-6), x(k-2 \times 6), \ldots, x(k-3 \times 6)]^T$ is composed of four values of the time series as done in [9]. The target output is the 84-step ahead
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].

\[
\text{NRMSE}_{84} = \sqrt{\frac{1}{N_r} \sum_{t=1}^{N_r} (z_t(84) - y_t(84))^2 / \sigma^2}.
\]

where $z_t(84)$ denotes the 84-step target value, $y_t(84)$ is the corresponding network prediction value, $\sigma^2$ 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 $[-6 \times 10^{-4}, 6 \times 10^{-4}]$. 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 $\alpha(k) = [y(k), y(k-10), y(k-2 \times 10), \ldots, y(k - 3 \times 10)]^T$ 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

<table>
<thead>
<tr>
<th>Method</th>
<th>Training time(s)</th>
<th>Testing NRMSE$_{84}$</th>
<th>Reservoir size</th>
<th>Spectral radius</th>
<th>Sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALM-ESN</td>
<td>151.21</td>
<td>$1.9268 \times 10^4$</td>
<td>200</td>
<td>0.8000</td>
<td>0.0200</td>
</tr>
<tr>
<td>OESN$^{[9]}$</td>
<td>163.49</td>
<td>$3.7925 \times 10^4$</td>
<td>400</td>
<td>0.9000</td>
<td>0.0350</td>
</tr>
<tr>
<td>SCR$^{[31]}$</td>
<td>123.51</td>
<td>$4.6548 \times 10^4$</td>
<td>300</td>
<td>0.8000</td>
<td>0.0033</td>
</tr>
<tr>
<td>DESN$^{[34]}$</td>
<td>156.62</td>
<td>$3.2968 \times 10^4$</td>
<td>400</td>
<td>0.9000</td>
<td>0.0237</td>
</tr>
<tr>
<td>RR-ESN$^{[18]}$</td>
<td>166.53</td>
<td>$2.8813 \times 10^4$</td>
<td>400</td>
<td>0.8500</td>
<td>0.0150</td>
</tr>
<tr>
<td>Lasso-ESN$^{[29]}$</td>
<td>-</td>
<td>$2.6532 \times 10^4$</td>
<td>400</td>
<td>0.8500</td>
<td>0.0150</td>
</tr>
<tr>
<td>BPSO-ESN$^{[22]}$</td>
<td>186.35</td>
<td>$2.3823 \times 10^4$</td>
<td>400</td>
<td>0.8500</td>
<td>0.0150</td>
</tr>
</tbody>
</table>
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.

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

\[
R(\theta) = \frac{\sum_{i=1}^{G} h(a_i - \theta)}{G}, \tag{28}
\]
$h(x) = \begin{cases} 
1 & x \leq 0 \\
0 & x > 0 
\end{cases}$  \hfill (29)
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 prediction demonstrate that the ALM-ESN exhibits better prediction performance than some existing ESN construction methods.

REFERENCES


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 Automation, 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.

***