Control and Optimization in Applied Mathematics (COAM)
DOI. 10.30473/coam.2020.51511.1137
Vol. 4, No. 1, Spring-Summer 2019 (83-101), ©2016 Payame Noor University, Iran

Stable Rough Extreme Learning Machines for the Identification of Uncertain Continuous-Time Nonlinear Systems

G. Ahmadi*
Department of Mathematics, Payame Noor University (PNU),
P.O. Box. 19395-3697, Tehran, Iran

Received: February 11, 2020; **Accepted:** October 5, 2020.

Abstract. Rough extreme learning machines (RELMs) are rough-neural networks with one hidden layer where the parameters between the inputs and hidden neurons are arbitrarily chosen and never updated. In this paper, we propose RELMs with a stable online learning algorithm for the identification of continuous-time nonlinear systems in the presence of noises and uncertainties, and we prove the global asymptotically convergence of the proposed learning algorithm using the Lyapunov stability theory. Then, we use the proposed methodology to identify the chaotic systems of Duffing's oscillator and Lorentz system. Simulation results show the efficiency of the proposed model.

Keywords. System identification, Extreme learning machine, Rough-neural network, Rough extreme learning machine, Lyapunov stability theory.

MSC. 93B30; 68T05.

* Corresponding author
g.ahmadi@pnu.ac.ir
http://mathco.journals.pnu.ac.ir

1 Introduction

System identification is an important field of science that focuses on organizing the knowledge about the systems, and it is necessary for many disciplines of applied sciences such as physics, economics, biology, and control engineering. In the literature, the dynamic systems are classified into two categories: discrete-time and continuous-time nonlinear systems (CTNSs). Most of the system identification techniques are developed for discrete-time nonlinear systems, and a few works are done for the identification of CTNSs [6, 7, 9, 27, 28, 32]. However, there are some reasons that continuous-time models are more reliable than discrete-time models. Most of the physical laws and classical theories such as Newton's laws, Faraday's laws, Laplace transforms, and PID controllers are created in the continuous-time description. During the transformation of continuous-time description of a system into the discrete-time description, loss of information may occur. The effects of noises on the discrete-time models are more than their effects on continuous-time models.

One of the most important approaches for identifying nonlinear systems is the employment of neural networks as identifiers [1, 18, 22, 23, 24]. The abilities of neural networks in system identification and control are shown during the last decades. However, there are a few works on the identification of CTNSs. Radial basis function networks have been used for the identification of CTNSs [22].

For increasing the abilities of neural networks to overcome the uncertainties in the applications, they are combined with the theories concerning the uncertainty. In this regard, based on rough set theory, Lingras proposed the rough-neural networks (R-NNs) [21]. Moreover, he defined the rough neuron as a pair of upper and lower bound neurons where the information exchanges between them. Rough-neural networks are used in different contexts successfully, such as predicting traffic amount, image processing, classification, wind speed prediction [5, 10, 11, 16, 17, 31]. Recently, Ahmadi et al. utilized R-NNs to identify discrete dynamic nonlinear systems [2, 3, 4].

Due to the existence of many parameters in the structure of R-NNs, their training is time-consuming. To overcome this problem, we propose the rough extreme learning machine (RELM) for the identification of CTNSs. RELM is an R-NN with one hidden layer where the parameters between the inputs and hidden neurons are arbitrarily chosen and never updated. In 2006, Huang et al. proposed an extreme learning machine (ELM) for the first time [12]. The training of ELMs is done significantly faster than the traditional feedforward neural networks. They have been used for different purposes in recent years, such as identification and control of nonlinear systems, classification, and so on [8, 18, 19, 30].

Recently, the authors published some works in the contexts of chaotic system identification using neural networks and R-NNs. Lamamra et al. utilized the multilayer perceptron to model the chaotic systems where the parameters are adjusted with a genetic algorithm [20]. Pan et al. employed the multilayer perceptron with a hybrid learning based on the genetic algorithm and the steepest descent method for the identification of chaotic systems [29]. Jahangir et al. used R-NNs to forecast the electricity price [17]. In their proposed model, the inputs are crisp, the rough neurons have only one particular output, and all of the parameters are adjusted using

the backpropagation algorithm. Feng et al. proposed the rough ELM for the classification of uncertain data [8]. In their model, the input data must be crisp, and the authors utilized the upper and lower approximation sets to adjust the parameters.

In this paper, we propose the RELM for the identification of CTNSs in the presence of noises. We adjust the parameters of RELM with an online Lyapunov-based learning algorithm, and we prove the global asymptotically convergence of this algorithm. This model admits interval data as input through their upper and lower bounds. We use the proposed methodology to identify the chaotic systems, Duffing's oscillator, and Lorentz system. Simulation results show the better performances of RELMs compared with that of ELMs.

We can summarize the contributions of this work as follows:

- In this work, we consider the problem of CTNS identification in the presence of uncertainties where, despite its importance, in the literature, there are a few works in this context.
- On the base of R-NNs, we propose a new structure of ELMs called RELM and utilize it for the identification of nonlinear systems.
- We propose an online Lyapunov-based learning algorithm for adjusting the trainable parameters of RELM.
- We utilize the proposed methodology for the identification of chaotic systems, Duffing's oscillator and Lorentz system in the presence of noises.

The paper is organized as follows. Section 2 gives the structure of RELM. In Section 3, we use the RELMs to identify the CTNSs. In Section 4, we utilize an online Lyapunov-based learning algorithm for training RELMs and prove the global asymptotically convergence of this algorithm. Section 5 carries out the simulation results, and Section 6 states the conclusions.

2 Structure of Rough Extreme Learning Machines

RELM is a rough neural network with one hidden layer where the parameters between the inputs and hidden neurons are randomly assigned and never updated. This Section gives the structure of RELM and computes the output vector. Consider the RELM where the hidden neurons are rough and the output neurons are conventional, as shown in Figure 1. We show the output vector of RELM with $\widehat{\mathbf{y}}$ and the input vector of RELM with

$$\mathbf{x} = [\overline{x}^1, \underline{x}^1, \overline{x}^2, \underline{x}^2, \ldots, \overline{x}^m, \underline{x}^m, 1]^T,$$

where $\underline{x}^i$ $(i \in \{1, 2, \ldots, m\})$ is the lower bound, and $\overline{x}^i$ $(i \in \{1, 2, \ldots, m\})$ is the upper bound of x^i. The component 1 in the vector $\mathbf{x}$ shows the input corresponding to the biases. Let $\underline{V}_r$ and $\overline{V}_r$ be the parameters between the inputs and lower bound neurons in the hidden layer and the parameters between the inputs and upper bound neurons in the hidden layer, respectively. According to the definition of RELM, $\underline{V}_r$ and $\overline{V}_r$ are some randomly chosen numbers and remain unchanged during the training process. Suppose that $\underline{W}$ and $\overline{W}$ be the parameters

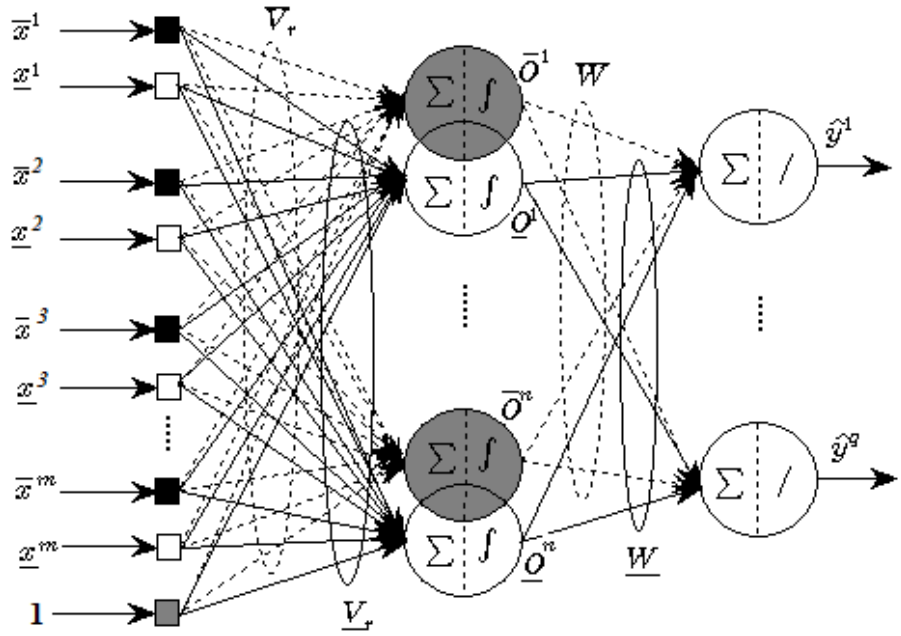

Figure 1: Structure of RELM.

between the hidden lower bound neurons and output neurons and the parameters between the hidden upper bound neurons and output neurons, respectively.

Further, let $\underline{I}$, $\overline{I}$, $\underline{O}$, and $\overline{O}$ be the inputs into the lower bound neurons in the hidden layer, the inputs into the upper bound neurons in the hidden layer, the outputs of lower bound neurons in the hidden layer, the outputs of upper bound neurons in the hidden layer, respectively. Besides, ϕ shows the activation function in the hidden layer. Then, we have

$$\begin{aligned} \underline{I} &= \underline{V}_r\mathbf{x}, \quad \overline{I} = \overline{V}_r\mathbf{x}, \\ \underline{O} &= \min\left(\phi(\underline{I}), \phi(\overline{I})\right), \quad \overline{O} = \max\left(\phi(\underline{I}), \phi(\overline{I})\right), \end{aligned}$$

and the output vector $\widehat{\mathbf{y}}$ of RELM is given by

$$\begin{aligned} \widehat{\mathbf{y}} &= \underline{WO} + \overline{WO} \\ &= \underline{W}\min\left(\phi(\underline{V}_r\mathbf{x}), \phi(\overline{V}_r\mathbf{x})\right) + \overline{W}\max\left(\phi(\underline{V}_r\mathbf{x}), \phi(\overline{V}_r\mathbf{x})\right). \end{aligned} \tag{1}$$

For simplicity, we use the following notations:

$$\underline{\phi} = \phi(\underline{V}_r\mathbf{x}), \quad \overline{\phi} = \phi(\overline{V}_r\mathbf{x}). \tag{2}$$

Therefore, we can write equation (1) as

$$\widehat{\mathbf{y}} = \underline{W}\min\left(\underline{\phi}, \overline{\phi}\right) + \overline{W}\max\left(\underline{\phi}, \overline{\phi}\right). \tag{3}$$

Remark 1. We can state the main differences of RELM with other ELMs as follows:

- RELM contains rough neurons in the hidden layer.

- Rough patterns can be used as the inputs of RELM through their upper and lower bounds.

These properties increase the capabilities of neural models to cope with the uncertainties and noises in the identification of nonlinear systems.

Remark 2. Recently, a particular type of ELM, called rough ELM is used to classify uncertain data [8]. The proposed structure and learning algorithm in the current paper is different from that work. In [8], the input data of the proposed model needs to be crisp. However, in this work, RELM admits rough patterns as inputs during the upper and lower bounds of features, which increases the flexibility of the model. Also, in [8], the upper and lower approximation sets are utilized to adjust the parameters. Nevertheless, this work uses a Lyapunov-based learning algorithm to train the RELM.

3 RELM as an Identifier

We can state a multi-input multi-output (MIMO) CTNS as follows:

$$\dot{\mathbf{z}} = f(\mathbf{z}, \mathbf{u}), \tag{4}$$

where $\mathbf{u}$ and $\mathbf{z}$ denote the system inputs and outputs, respectively. Suppose that (4) is completely controllable and observable, and the solution of (4) is existed and unique. In the presence of noises, the components of $\mathbf{z}$ are uncertain. Suppose that $\underline{\mathbf{z}}$ and $\overline{\mathbf{z}}$ are the lower and upper bounds for $\mathbf{z}$, respectively. Besides, $\underline{\mathbf{u}}$ and $\overline{\mathbf{u}}$ show the lower and upper bounds for $\mathbf{u}$, respectively.

The system (4) can be restated as

$$\dot{\mathbf{z}} = A\mathbf{z} + g(\mathbf{z}, \mathbf{u}), \tag{5}$$

where we represent the system nonlinearity with $g(\mathbf{z}, \mathbf{u}) = f(\mathbf{z}, \mathbf{u}) - A\mathbf{z}$, and A is a Hurwitz matrix. Suppose that the RELM can model g in equation (5), using the parameters $\underline{W}_\star$ and $\overline{W}_\star$ with an accuracy of ϵ. Therefore, using equation (3), we can write

$$\dot{\mathbf{z}} = A\mathbf{z} + \underline{W}_\star \min\left(\underline{\phi}, \overline{\phi}\right) + \overline{W}_\star \max\left(\underline{\phi}, \overline{\phi}\right) + \epsilon, \tag{6}$$

where $\mathbf{x}$ contains the upper and lower bounds of inputs and states of RELMs. Then, we can construct the parametric model of the system (4) as

$$\dot{\widehat{\mathbf{z}}} = A\widehat{\mathbf{z}} + \underline{\widehat{W}} \min\left(\underline{\phi}, \overline{\phi}\right) + \overline{\widehat{W}} \max\left(\underline{\phi}, \overline{\phi}\right), \tag{7}$$

where $\underline{\widehat{W}}$ and $\overline{\widehat{W}}$ represent the parameters to estimate $\underline{W}_\star$ and $\overline{W}_\star$, respectively.

Remark 3. According to the universal approximation theorems, the neural networks can approximate the continuous functions with arbitrary precision [14]. Also, universal approximation has been proved for ELMs [13]. On the base of these theories, in this work, we suppose that RELM can model the function g with the unknown ideal parameters $\underline{W}_\star$ and $\overline{W}_\star$. Therefore, in the applications, a learning algorithm is used to estimate these parameters.

Using equations (6) and (7), we have

$$\begin{aligned}\dot{\mathbf{e}} &= \dot{\mathbf{z}} - \dot{\hat{\mathbf{z}}} \\ &= A\mathbf{e} + \underline{\widetilde{W}} \min\left(\underline{\phi}, \overline{\phi}\right) + \overline{\widetilde{W}} \max\left(\underline{\phi}, \overline{\phi}\right) + \epsilon, \end{aligned} \tag{8}$$

where

$$\underline{\widetilde{W}} = \underline{W}_\star - \underline{\widehat{W}}, \quad \overline{\widetilde{W}} = \overline{W}_\star - \overline{\widehat{W}}. \tag{9}$$

4 Lyapunov-Based Learning Algorithm for RELMs

In this Section, based on the Lyapunov stability theory, an online learning algorithm is proposed for RELM. In the proof of Theorem 1, the definition of function spaces $\mathbf{L}_2$ and $\mathbf{L}_\infty$ would be needed [15]. For continuous function f, the $\mathbf{L}_2$ norm is defined as follow:

$$\|f\|_2 = \left(\int_0^\infty |f(t)|^2 dt\right)^{1/2}. \tag{10}$$

If $\|f\|_2$ exists, then we say $f \in \mathbf{L}_2$. The $\mathbf{L}_\infty$ norm is defined as follow:

$$\|f\|_\infty = \sup_{t\geq 0} |f(t)|. \tag{11}$$

If $\|f\|_\infty$ exists, then we say $f \in \mathbf{L}_\infty$.

Theorem 1. Suppose that the model RELM, which is given by (7), is utilized for the identification of the system (4) and it is trained by the following laws:

$$\dot{\underline{\widehat{W}}} = \mathbf{e}\left[\min(\underline{\phi}, \overline{\phi})\right]^T \Gamma_1^{-1}, \tag{12}$$

$$\dot{\overline{\widehat{W}}} = \mathbf{e}\left[\max(\underline{\phi}, \overline{\phi})\right]^T \Gamma_2^{-1}, \tag{13}$$

where the symmetric matrices Γ_1 and Γ_2 are the gains of learning. Assume that

$$\|\mathbf{e}\| \geq \frac{\|\epsilon\|}{|\lambda_{min}(A)|}, \tag{14}$$

where $\lambda_{\min}(A)$ is the eigenvalue of A with the least absolute value. In this case, the modeling error $\mathbf{e}$ converges to zero. Besides, the weights $\underline{\widehat{W}}, \overline{\widehat{W}}$, and the predictions $\widehat{\mathbf{z}}$ are bounded.

Proof. Let

$$v = \frac{1}{2}\mathbf{e}^T\mathbf{e} + \frac{1}{2}\mathrm{tr}\left(\underline{\widetilde{W}}\Gamma_1\underline{\widetilde{W}}^T\right) + \frac{1}{2}\mathrm{tr}\left(\overline{\widetilde{W}}\Gamma_2\overline{\widetilde{W}}^T\right), \tag{15}$$

where Γ_1 and Γ_2 represent the gains of learning. Then, according to (8), we have

$$\dot{v} = \mathbf{e}^T\dot{\mathbf{e}} + \mathrm{tr}\left(\dot{\underline{\widetilde{W}}}\Gamma_1\underline{\widetilde{W}}^T\right) + \mathrm{tr}\left(\dot{\overline{\widetilde{W}}}\Gamma_2\overline{\widetilde{W}}^T\right)$$

$$
\begin{aligned}
&= \mathbf{e}^T A\mathbf{e} + \mathbf{e}^T \widetilde{\underline{W}} \min\left(\underline{\phi}, \overline{\phi}\right) + \mathbf{e}^T \widetilde{\overline{W}} \max\left(\underline{\phi}, \overline{\phi}\right) + \epsilon + \mathrm{tr}\left(\dot{\widetilde{\underline{W}}} \Gamma_1 \widetilde{\underline{W}}^T\right) \\
&\quad + \mathrm{tr}\left(\dot{\widetilde{\overline{W}}} \Gamma_2 \widetilde{\overline{W}}^T\right) \\
&= \mathbf{e}^T A\mathbf{e} + \mathbf{e}^T \epsilon + \mathrm{tr}\left(\mathbf{e}\left[\min\left(\underline{\phi}, \overline{\phi}\right)\right]^T \widetilde{\underline{W}}^T\right) + \mathrm{tr}\left(\mathbf{e}\left[\max\left(\underline{\phi}, \overline{\phi}\right)\right]^T \widetilde{\overline{W}}^T\right) \\
&\quad + \mathrm{tr}\left(\dot{\widetilde{\underline{W}}} \Gamma_1 \widetilde{\underline{W}}^T\right) + \mathrm{tr}\left(\dot{\widetilde{\overline{W}}} \Gamma_2 \widetilde{\overline{W}}^T\right) \\
&= \mathbf{e}^T A\mathbf{e} + \mathbf{e}^T \epsilon + \mathrm{tr}\left(\mathbf{e}\left[\min\left(\underline{\phi}, \overline{\phi}\right)\right]^T \widetilde{\underline{W}}^T + \dot{\widetilde{\underline{W}}} \Gamma_1 \widetilde{\underline{W}}^T\right) \\
&\quad + \mathrm{tr}\left(\mathbf{e}\left[\max\left(\underline{\phi}, \overline{\phi}\right)\right]^T \widetilde{\overline{W}}^T + \dot{\widetilde{\overline{W}}} \Gamma_2 \widetilde{\overline{W}}^T\right). \qquad (16)
\end{aligned}
$$

From the fact that

$$
\widetilde{\underline{W}} = \underline{W}_\star - \widehat{\underline{W}}, \ \widetilde{\overline{W}} = \overline{W}_\star - \widehat{\overline{W}}, \qquad (17)
$$

we have

$$
\dot{\widetilde{\underline{W}}} = -\dot{\widehat{\underline{W}}}, \ \ \dot{\widetilde{\overline{W}}} = -\dot{\widehat{\overline{W}}}. \qquad (18)
$$

Therefore

$$
\begin{aligned}
\dot{v} &= \mathbf{e}^T A\mathbf{e} + \mathbf{e}^T \epsilon + \mathrm{tr}\left(\mathbf{e}\left[\min(\underline{\phi}, \overline{\phi})\right]^T \widetilde{\underline{W}}^T - \right. \\
&\quad \left. \dot{\widehat{\underline{W}}} \Gamma_1 \widetilde{\underline{W}}^T\right) + \mathrm{tr}\left(\mathbf{e}\left[\max(\underline{\phi}, \overline{\phi})\right]^T \widetilde{\overline{W}}^T - \dot{\widehat{\overline{W}}} \Gamma_2 \widetilde{\overline{W}}^T\right) \\
&= \mathbf{e}^T A\mathbf{e} + \mathbf{e}^T.\epsilon \qquad (19)
\end{aligned}
$$

In (19), the last equality is the result of assumptions (12) and (13). Now, we have

$$
\begin{aligned}
\dot{v} &= \mathbf{e}^T A\mathbf{e} + \mathbf{e}^T \epsilon \\
&\leq -\|\mathbf{e}\|^2 |\lambda_{\min}(A)| + \|\mathbf{e}\| \|\epsilon\|. \qquad (20)
\end{aligned}
$$

However, the expression (20) is a polynomial of degree 2 in $\|\mathbf{e}\|$ and according to (14), we have $\dot{v} < 0$. Therefore, v is a decreasing function and for every $t > 0$, we have $v < v(0)$. Due to the positive definiteness of v, for every $t > 0$ we have $v > 0$. Then, for every $t > 0$, $0 < v < v(0)$ and $v \in \mathbf{L}_\infty$. Also, $\mathbf{e} \in \mathbf{L}_\infty$. From (20), we have

$$
\begin{aligned}
0 &< \int_0^\infty \|\mathbf{e}\|^2 \lambda_{\min}(A) dt - \int_0^\infty \|\mathbf{e}\| \|\epsilon\| dt \\
&\leq -\int_0^\infty \dot{v} dt = v(0) - v(\infty) < \infty, \qquad (21)
\end{aligned}
$$

that implies $\mathbf{e} \in \mathbf{L}_2$. Using these results, $\mathbf{e} \in \mathbf{L}_\infty \cap \mathbf{L}_2$. Then, by employing the Barbalat's lemma [15], we have $\mathbf{e} \longrightarrow 0$ as $t \longrightarrow \infty$. Therefore, according to (12) and (13),

$$
\dot{\widehat{\underline{W}}}, \dot{\widehat{\overline{W}}} \longrightarrow 0. \qquad (22)
$$

Moreover, $\widehat{\underline{W}}$ and $\widehat{\overline{W}}$ are bounded. □

As a result, the modeling error converges to zero, and the weights and predictions are bounded.

5 Simulation Results

This Section uses RELMs with the learning laws (12) and (13) to identify chaotic systems Duffing's oscillator and Lorentz system with noises, and we compare their performances with that of ELMs. A chaotic system is a dynamic system that is very sensitive to initial conditions. The identification of chaotic systems is an active field of research [26, 29, 20]. Chaotic systems have many applications in different fields of science and engineering. We use the NARX configuration in these models. Simulation results show a better performances of RELMs compared with that of ELMs.

When the state $\mathbf{z}$ of the nonlinear dynamic system is noisy, to implement the RELMs, the upper and lower bounds $\overline{\mathbf{z}}$ and $\underline{\mathbf{z}}$ are required. In these simulations, to approximate $\overline{\mathbf{z}}$ and $\underline{\mathbf{z}}$, we repeat the adding noise with a specified signal-to-noise ratio (SNR) to the data set multiple times. The performance metric is the identification mean squared errors (MSEs). For implementing this methodology on digital computers, we use the Runge-Kutta method of order four.

5.1 Duffing's Oscillator

Consider the following chaotic system called Duffing's equation:

$$\ddot{x} = -p\dot{x} - p_1 x - p_2 x^3 + u. \tag{23}$$

Duffing's equation describes a specific nonlinear circuit, observed in many mechanical problems. Here, t is the time variable, p, p_1, p_2 and q are real constants, and u is the control input $u = q\cos(\omega t)$ where ω is the frequency.We can write it as

$$\begin{cases} \dot{z}_1 &= z_2, \\ \dot{z}_2 &= -pz_2 - p_1 z_1 - p_2 z_1^3 + u. \end{cases} \tag{24}$$

For this simulation, $p = 0.4$, $p_1 = 1.1$, $p_2 = 1$, $\omega = 1.8$, $q = 1.65$ and $z_1(0) = z_2(0) = 0$ are considered. The system (24) with noise (SNR=15 dB) is identified by ELM and RELM. In their hidden layer, we use the hyperbolic tangent as activation function. The initial values of trainable parameters $\widehat{W}$, $\underline{\widehat{W}}$ and $\overline{\widehat{W}}$ are some random numbers between -1 and 1. The constant parameters V_r, $\underline{V}_r$ and $\overline{V}_r$ are random numbers between -2 and 2. The input vector of ELM is $x = [u, z_1, z_2, 1]^T$ and the input vector of RELM is

$$x = \left[u, \overline{z}_1, \underline{z}_1, \overline{z}_2, \underline{z}_2, 1\right]^T.$$

We choose the design matrix A as

$$A = \begin{bmatrix} -5 & 0 \\ 0 & -5 \end{bmatrix}. \tag{25}$$

Remark 4. According to the presented theory, the matrix A must be Hurwitz (every eigenvalue of A has strictly negative real part). In the simulation results, we choose the matrix A empirically, and it is possible to choose a nondiagonal matrix.

Table 1: Performances comparison of ELM and RELM models in the identification of noisy Duffing's Oscillator (SNR=15 dB). We show the number of (rough) neurons in the hidden layer, in column n_h.

Model	n_h	Parameters	Train MSE	Test MSE
ELM	20	40	0.0036	0.0010
ELM	30	60	0.0023	0.0008
ELM	40	80	0.0014	0.0008
ELM	50	100	0.0013	0.0008
ELM	70	140	0.0014	0.0007
RELM	10	40	0.0014	0.0003
RELM	15	60	0.0010	0.0002
RELM	20	80	0.0009	0.0002
RELM	25	100	0.0007	0.0001
RELM	35	140	0.0007	0.0001

The design parameters of the learning algorithm for ELM are selected as follows:

$$n_h = 20,\ 30,\ 40,\ 50,\ 70,\quad \Gamma_1 = 100 I_{n_h \times n_h}, \tag{26}$$

where Γ_1 denotes the learning rates. The design parameters of the learning algorithm for RELM are selected as follows:

$$n_h = 10,\ 15,\ 20,\ 25,\ 35,\quad \Gamma_1 = 100 I_{n_h \times n_h},\quad \Gamma_2 = 100 I_{n_h \times n_h}, \tag{27}$$

where Γ_1 and Γ_2 denote the learning rates. It must be mentioned that for generating the data set and the simulation aspects of the proposed methodology, solving the differential equations by one of the numerical methods are necessary. Here, we use the Runge-Kutta method of order four. We choose the sampling time for this simulation as 0.01.

In identifying (24), we list the training and testing MSEs of ELM and RELM models in Table 1. In this Table, the models with an equal number of trainable parameters are comparable. Due to the number of parameters, we can compare the performances of ELM with 20, 30, 40, 50, and 70 hidden neurons with the performances of RELM with 10, 15, 20, 25, and 35 hidden rough neurons, respectively. For example, the training MSE of ELM with 50 hidden neurons is 0.0013, and the testing MSE is 0.0008, where the training MSE of RELM with 25 hidden rough neurons is 0.0007, and the testing MSE is 0.0001.

Figure 2 shows the parameter evolution and error convergence in the training of ELM with 50 hidden neurons for the noisy Duffing's oscillator (SNR=15 dB). Figure 3 show the parameter evolution and error convergence in the training of RELM with 25 hidden rough neurons for the noisy Duffing's oscillator (SNR=15 dB).

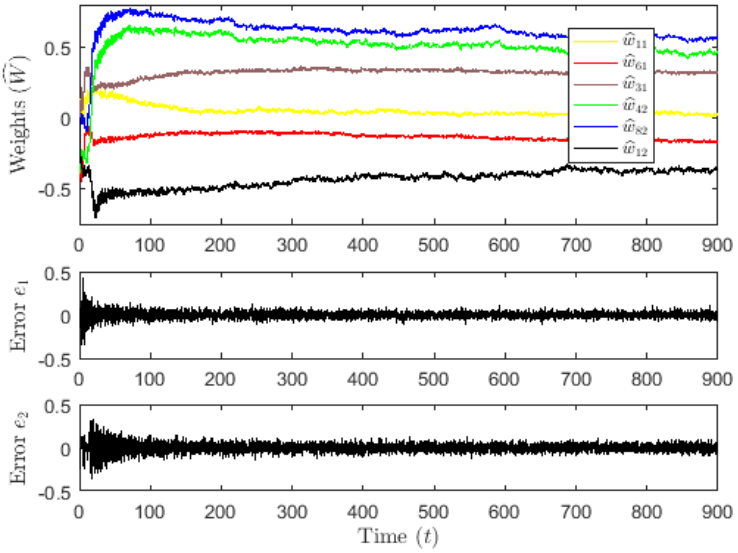

Figure 2: Parameter evolution and error convergence in the training of ELM with Fifty neurons in the hidden layer, in the identification of noisy Duffing's oscillator (SNR=15 dB).

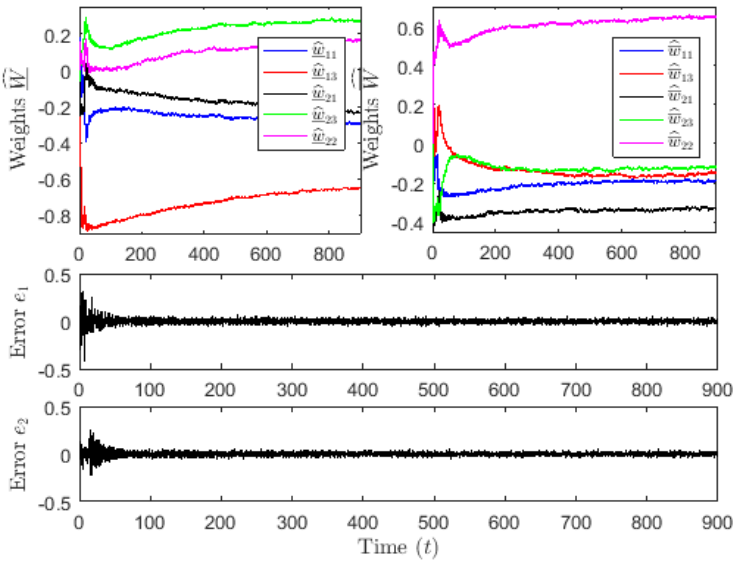

Figure 3: Parameter evolution and error convergence in the training of RELM with Twenty-five hidden rough neurons, in the identification of noisy Duffing's oscillator (SNR=15 dB).

The colored line graphs in these figures show some of the parameters (weights) in the models which are chosen randomly. Figures 4 and 5 show the true states, the estimated states, and the errors in the testing of ELM with 50 neurons in the hidden layer and RELM with 25 rough neurons in the hidden layer, in the identification of noisy Duffing's oscillator (SNR=15 dB).

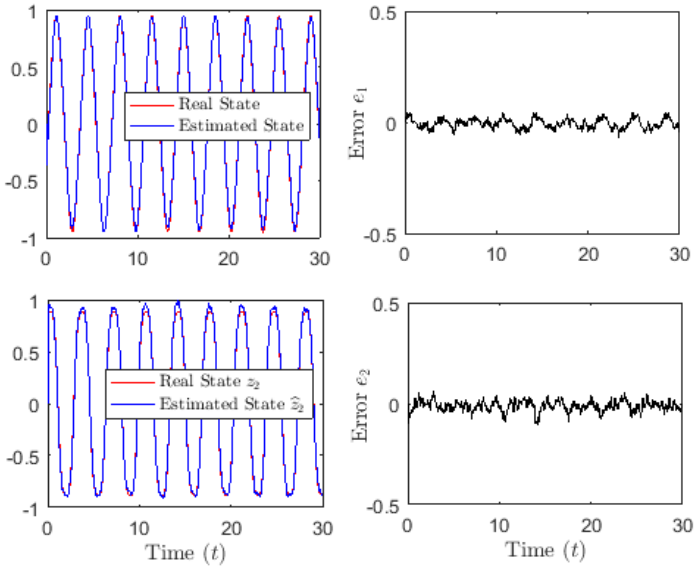

Figure 4: The actual states, the estimated states, and the errors in the testing of ELM with Fifty hidden neurons, in the identification of noisy Duffing's oscillator (SNR=15 dB).

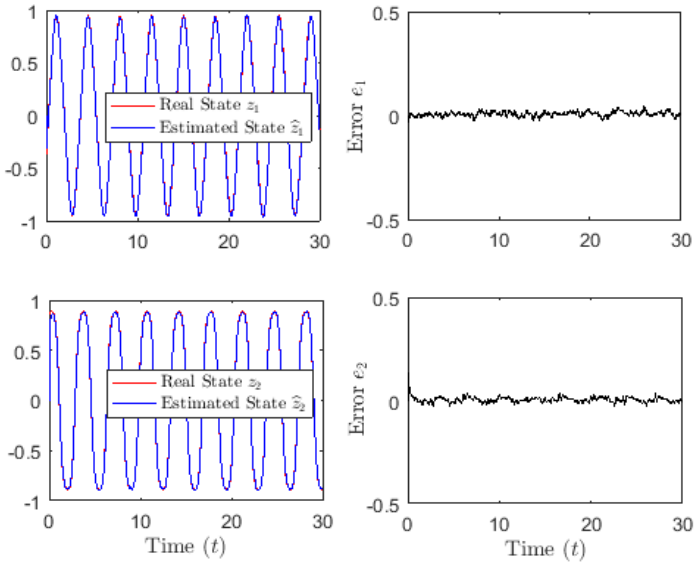

Figure 5: The actual states, the estimated states, and the errors in the testing of RELM with Twenty-five hidden rough neurons, in the identification of noisy Duffing's oscillator (SNR=15 dB).

From Table 1 and Figures 2-5, we can conclude that in the presence of noise, the performance of RELM in the identification of (28) is better than that of ELM. According to Table 1, increasing the number of hidden neurons generally results in decreasing the identification error. Therefore, to achieve a better model, we can increase the number of hidden neurons.

5.2 Lorentz System

The dynamic equations of Lorentz oscillator are as follows:

$$\begin{cases} \dot{z}_1 &= \sigma(z_2 - z_1), \\ \dot{z}_2 &= r z_1 - z_2 - z_1 z_3, \\ \dot{z}_3 &= z_1 z_2 - b z_3, \end{cases} \tag{28}$$

where $\sigma,\ r,\ b > 0$ are system parameters. For this simulation, $\sigma = 10,\ r = 28,\ b = 8/3$ and $z_1(0) = 7.5,\ z_2(0) = -9,\ z_3(0) = 28$ are considered. The identification of noisy Lorentz oscillator (28) where SNR=15 dB, is done by ELM and RELM. In the hidden layer of these models, the hyperbolic tangent is used as an activation function. In this simulation, all data is normalized to be in the interval $[-1, 1]$. The initial values of trainable parameters $\widehat{W}$, $\underline{\widehat{W}}$, and $\overline{\widehat{W}}$ are some random numbers between -1 and 1. The constant parameters V_r, $\underline{V}_r$, and $\overline{V}_r$ are random numbers between -1 and 1. The input vector of ELM is $x = [z_1, z_2, z_3, 1]^T$ and the input vector of RELM is

$$x = \left[\overline{z}_1, \underline{z}_1, \overline{z}_2, \underline{z}_2, \overline{z}_3, \underline{z}_3, 1\right]^T.$$

We choose the design matrix A as

$$A = \begin{bmatrix} -60 & 0 & 0 \\ 0 & -60 & 0 \\ 0 & 0 & -120 \end{bmatrix}. \tag{29}$$

We select the design parameters of the learning algorithm for ELM as

$$n_h = 20,\ 30,\ 40,\ 50,\ 80,\ 100, \quad \Gamma_1 = 20 I_{n_h \times n_h}, \tag{30}$$

where Γ_1 denotes the learning rates. We select the design parameters of the learning algorithm for RELM as

$$n_h = 10,\ 15,\ 20,\ 25,\ 40,\ 50,\ \ \Gamma_1 = 20 I_{n_h \times n_h}, \quad \Gamma_2 = 20 I_{n_h \times n_h}, \tag{31}$$

where Γ_1 and Γ_2 denote the learning rates. It must be mentioned that for generating the data set and the simulation aspects of the proposed methodology, solving the differential equations by one of the numerical methods are necessary. Here, we use the Runge-Kutta method of order four. We choose the sampling time for this simulation as 0.01.

We list the MSEs of identifying (28) with ELM and RELM models in training and testing in Table 2. In this Table, the models with an equal number of trainable parameters are comparable. Due to the number of parameters, the performance of ELM with 20, 30, 40, 50, 80, and 100 neurons in the hidden layer are comparable with the performance of RELM with 10, 15, 20, 25, 40, and 50 rough neurons in the hidden layer, respectively. For example, the training MSE of ELM with 50 hidden neurons is 0.0025, and the testing MSE is 0.0012, where the training MSE of RELM with 25 hidden rough neurons is 0.0013, and the testing MSE is 0.0003.

Figures 6 and 7 show the parameter evolution and error convergence in the training of ELM with 50 neurons in the hidden layer and RELM with 25 rough neurons in the hidden layer, for the same data set of noisy Lorentz system (SNR=15 dB), respectively. The colored line graphs

Table 2: Performances comparison of ELM and RELM models in the identification of noisy Lorentz system (SNR=15 dB). The number of (rough) neurons in the hidden layer is shown in column n_h. The MSEs are normalized.

Model	n_h	Parameters	Train MSE	Test MSE
ELM	20	60	0.0039	0.0015
ELM	30	90	0.0028	0.0013
ELM	40	120	0.0025	0.0013
ELM	50	150	0.0025	0.0012
ELM	80	240	0.0018	0.0011
ELM	100	300	0.0015	0.0010
RELM	10	60	0.0055	0.0014
RELM	15	90	0.0038	0.0009
RELM	20	120	0.0016	0.0004
RELM	25	150	0.0013	0.0003
RELM	40	240	0.0009	0.0002
RELM	50	300	0.0007	0.0002

in these figures show some of the parameters (weights) in the models which are chosen randomly. Figures 8 and 9 show the actual states and the estimated states in the testing of ELM with 50 hidden neurons and RELM with 25 hidden rough neurons, in the identification of noisy Lorentz system (SNR=15 dB), respectively. Also, Figure 10 shows the parameter evolution and error convergence in the training of RELM with 25 hidden rough neurons in the identification of the Lorentz system (28) without noises. This Figure confirms the modeling error convergence to zero as it is shown in Theorem 1. Nevertheless, in Figures 2, 3, 6, and 7, the noises influence the convergence of the identification error to zero. According to Table 2, increasing the number of hidden neurons generally results in decreasing the identification error. Therefore, to achieve a better model, we can increase the number of hidden neurons.

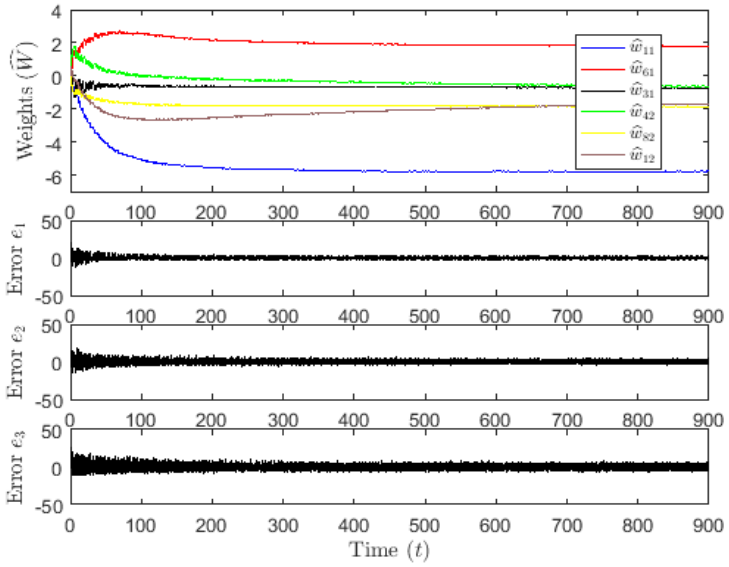

Figure 6: Parameter evolution and error convergence in the training of ELM with Fifty hidden neurons, in the identification of noisy Lorentz system (SNR=15 dB).

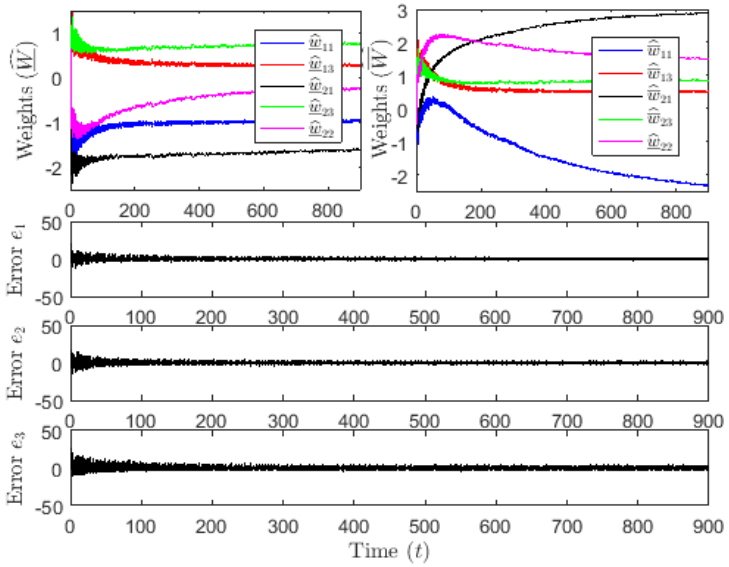

Figure 7: Parameter evolution and error convergence in the training of RELM with Twenty-five hidden rough neurons, in the identification of noisy Lorentz system (SNR=15 dB).

From Table 2 and Figures 6-9, we can conclude that the performance of RELM in identifying (28) with noise is better than that of ELM. According to Table 2, increasing the number of hidden neurons generally results in decreasing the identification error. Therefore, to achieve

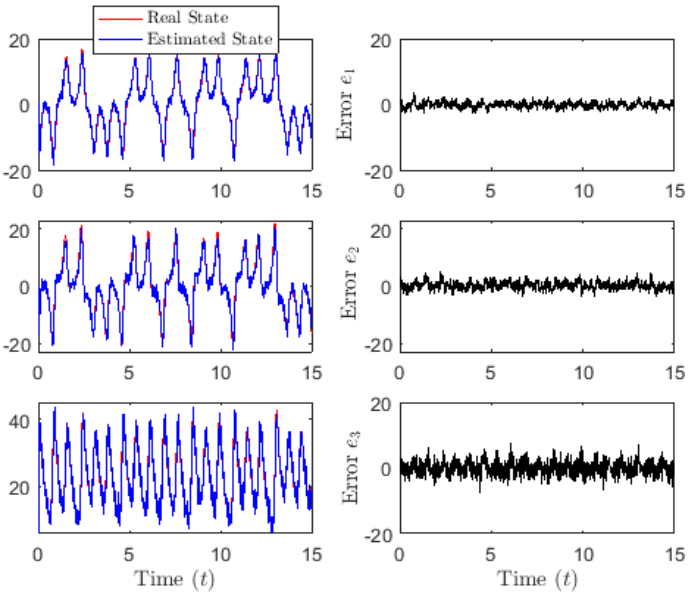

Figure 8: The actual states, the estimated states, and the errors in the testing of ELM with Fifty hidden neurons, in the identification of noisy Lorentz system (SNR=15 dB).

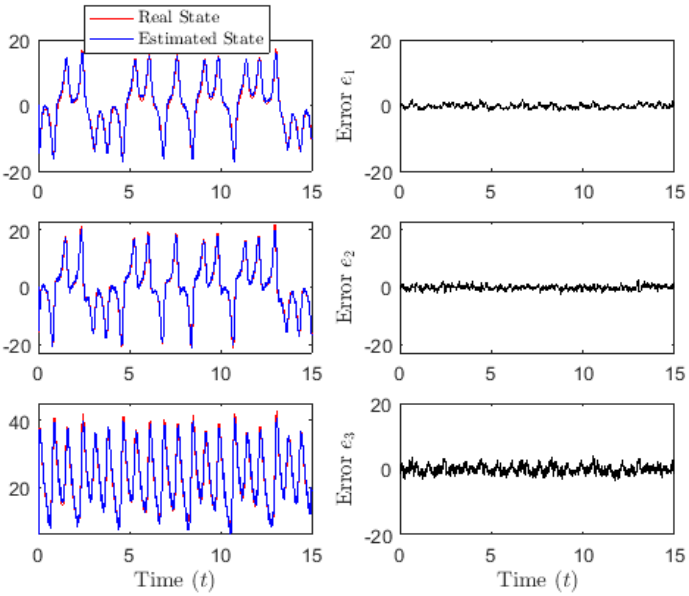

Figure 9: The actual states, the estimated states, and the errors in the testing of RELM with Twenty-five hidden rough neurons, in the identification of noisy Lorentz system (SNR=15 dB).

a better model we can increase the number of hidden neurons. Moreover, as a result, the identification error tends to zero.

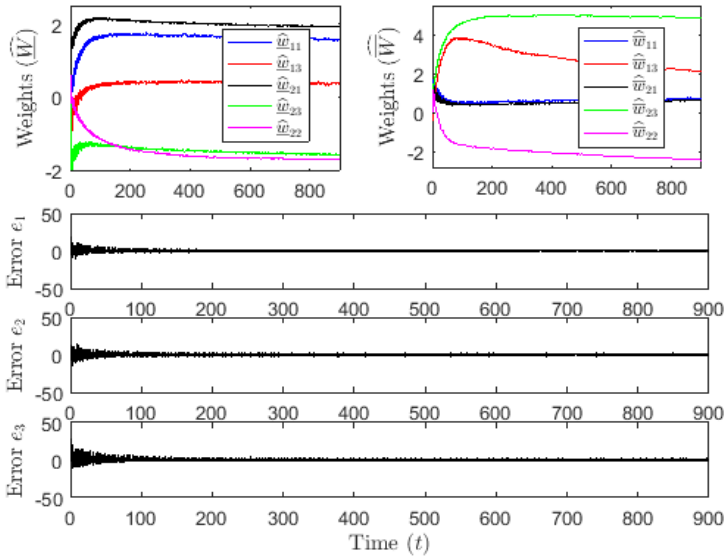

Figure 10: The parameter evolution and error convergence in the training of RELM with Twenty-five hidden rough neurons in the identification of the Lorentz system without noises.

6 Conclusion

Rough-neural networks (R-NNs) are some powerful models in dealing with the uncertainty in the identification of nonlinear systems. Nevertheless, the training of R-NNs is time-consuming. In this paper, to overcome the problem mentioned above, we used the extreme learning machines (RELMs) to identify the continuous-time nonlinear systems with noises in a series-parallel configuration. The trainable parameters of RELMs are less than R-NNs, and therefore, the training process takes less time. Based on the Lyapunov stability theory, we developed an online parameter adjustment algorithm to train the RELM. In the simulation results, the noise reduction ability and the fast error convergence of RELMs against the ELMs are shown. Our future work focuses on designing the controllers for dynamic nonlinear systems based on RELMs.

References

[1] Abdollahi F., Talebi A., Patel R. (2006). "Stable identification of nonlinear systems using neural networks: Theory and experiments", IEEE/ASME Transactions on Mechatronics, 11(4), 488-495.

[2] Ahmadi G., Teshnehlab M. (2016). "Designing and implementation of stable sinusoidal rough-neural identifier", IEEE Transactions on Neural Networks and Learning Systems, 28(8), 1774-1786.

[3] Ahmadi G., Teshnehlab M. (2020). "Identification of multiple input-multiple output nonlinear system cement rotary kiln using stochastic gradient-based rough-neural network", Journal of AI and Data Mining, 8(3), 417-425.

[4] Ahmadi G., Teshnehlab M., Soltanian F. (2018). "A Higher Order Online Lyapunov-Based Emotional Learning for Rough-Neural Identifiers", Control and Optimization in Applied Mathematics (COAM), 3(1), 87-108.

[5] Alehasher S., Teshnehlab M. (2012). "Implementation of rough neural networks with probabilistic learning for nonlinear system identification", J. Control 6(1), 41-50.

[6] Cellier F. E. (1991). "Continuous system modelling", Springer-Verlag, New York.

[7] Coca D., Billings S. (1997). "Continuous-time system identification for linear and nonlinear systems using wavelet decompositions", International Journal of Biforcation and Chaos 7(1), 87-96.

[8] Feng L., Xu S., Wang F., Liu S., Qiao H. (2019). "Rough extreme learning machine: A new classification method based on uncertainty measure", Neurocomputing, 325, 269-282.

[9] Garnier H., Wang L. (2008). "Identification of continuous-time models from sampled data", Springer-Verlag, London.

[10] Hassan Y. (2014). "Rough neural networks in adapting cellular automata rule for reducing image noise", International Journal of Computer, Information, Systems and Control Engineering, 8(1), 71-74.

[11] Hassanien A., Slezak D. (2006). "Rough-neural intelligent approach for image classification: A case of patients with suspected breast cancer", International Journal of Hybrid Intelligent Systems, 3, 205-218.

[12] Huang G. B., Zhu Q.-Y., Siew C.-K. (2006). "Extreme learning machine: Theory and applications", Neurocomputing, 70, 489-501.

[13] Huang G. B., Zhou H., Ding X., Zhang R. (2012). "Extreme learning machine for regression and multiclass classification", IEEE Transaction on Systems, Man, and Cybernetics—Part B: Cybernetics, 42(2), 513-529.

[14] Hykin S. (1998). "Neural networks: A comprehensive foundation", Prentice Hall International, Canada.

[15] Ioannou P., Sun J. (1996). "Robust adaptive control", Prentice Hall, New Jersey.

[16] Jahangir H., Golkar M. A., Alhameli F., Mazouz A., Ahmadian A., Elkamel A. (2020). "Short-term wind speed forecasting framework based on stacked denoising auto-encoders with rough ANN", Sustainable Energy Technologies and Assessments, 38, 100601.

[17] Jahangir H., Tayarani H., Baghali S., Ahmadian A., Elkamel A., Golkar M. A. (2020). "A novel electricity price forecasting approach based on dimension reduction strategy and rough artificial neural networks", IEEE Transactions on Industrial Informatics, 16(4), 2369-2381.

[18] Janakiraman V. M., Assanis D. (2012). "Lyapunov method based online identification of nonlinear systems using extreme learning machines", Computing Research Repository (CoRR):1211.1441, pp 1-8.

[19] Janakiraman V. M., Nguyen X., Assanis D., (2016). "Stochastic gradient based extreme learning machines for stable online learning of advanced combustion engines", Neurocomputing, 177, 304-316.

[20] Lamamra K., Vaidyanathan S., Azar A. T., Ben Salah C. (2017). "Chaotic system modelling using a neural network with optimized structure", in: Azar A. T. et al. (eds.), Fractional order control and synchronization of chaotic systems, Studies in Computational Intelligence 688, Springer International Publishing AG.

[21] Lingras P. (1996). "Rough neural networks", in: Proceedings of the 6th international conference on information processing and management of uncertainty (IPMU), Granada, 1445-1450.

[22] Liu G., Kadirkamanathan V., Billings S. (1994). "Stable sequential identification of continuous nonlinear dynamical systems by growing RBF networks", PhD thesis, Research Report No. 547, Depatment of Automatic Control and System Engineering, University of Sheffild, UK.

[23] Narendra K., Parthasarathy K. (1990). "Identification and control of dynamical systems using neural networks", IEEE Trans. Neural Networks, 1(1), 4-27.

[24] Nelles O. (2001). "Nonlinear system identification: From classical approaches to neural networks and fuzzy models", Springer-Verlag, Berlin.

[25] Pawlack Z. (1982). "Rough sets", International Journal of Computer and Information Sciences, 11(5), 341-356.

[26] Poznyak A. S., Yu W., Sanchez E. N. (1982). "Identification and control of unknown chaotic systems via dynamic neural networks", IEEE Transactions on Circuits and Systems—I: Fundamental Theories and Applications, 46(12).

[27] Rao G., Unbehauen H. (2006). "Identification of continuous-time systems", IEE Proc.-Control Theory Appl., 153(2), 185-220.

[28] Ren X., Rad A., Chan P., Lo W. (2003). "Identification and control of continuous-time nonlinear systems via dynamic neural networks", IEEE Transactions on Industrial Electronics, 50(3), 478-486.

[29] Pan S. T., Lai C. C. (2008). "Identification of chaotic systems by neural network with hybrid learning algorithm", Chaos, Solitons and Fractals, 37, 233-244.

[30] Wang Z., Li M., Wang H., Jiang H., Yao Y., Zhang H., Xin J. (2019). "Breast cancer detection using extreme learning machine based on feature fusion with CNN deep features", IEEE Access, 105146-105158.

[31] Yamaguchi D., Katayama F., Takahashi M., Arai M., Mackin K. (2008). "The medical diagnostic support system using extended rough neural network and multiagent", Artificial Life and Robotics, 13(1), 184-187.

[32] Zhang B., Billings S. (2015). "Identification of continuous-time non- linear systems: The nonlinear difference equation with moving average noise (ndema) framework", Mechanical Systems and Signal Processing, 60, 810-835.

ماشین‌های یادگیری حدی راف پایدار برای شناسایی سیستم‌های غیرخطی زمان-پیوسته غیرقطعی

احمدی، ق.
ایران، تهران، دانشگاه پیام‌نور، گروه ریاضی، صندوق پستی ۱۹۳۹۵-۳۶۹۷
g.ahmadi@pnu.ac.ir

تاریخ دریافت: ۲۲ بهمن ۱۳۹۸ تاریخ پذیرش: ۱۴ مهر ۱۳۹۹

چکیده

ماشین‌های یادگیری حدی راف شبکه‌های راف-عصبی با یک لایه پنهان هستند که در آنها پارامترهای بین ورودی‌ها و نورون‌های پنهان به صورت تصادفی انتخاب می‌شوند و هرگز به روز نمی‌شوند. در این مقاله، ماشینهای یادگیری حدی راف با یک الگوریتم یادگیری برخط پایدار را برای شناسایی سیستمهای غیرخطی زمان-پیوسته در حضور نویزها و عدم‌قطعیت‌ها پیشنهاد می‌کنیم و با استفاده از نظریه پایداری لیاپانوف، همگرایی مجانبی سراسری الگوریتم یادگیری پیشنهاد شده را اثبات می‌کنیم. سپس، از روش پیشنهاد شده برای شناسایی سیستم‌های آشوبی نوسانگر دافینگ و سیستم لورنز بهره می‌گیریم. نتایج شبیه‌سازی کارآمدی مدل پیشنهادی را نشان می‌دهد.

کلمات کلیدی

شناسایی سیستم، ماشین یادگیری حدی، شبکه راف-عصبی، ماشین یادگیری حدی راف، نظریه پایداری لیاپانوف.