Chaotic diagonal recurrent neural network*

Wang Xing-Yuan(王兴元) and Zhang Yi(张 诣)†

School of Electronic and Information Engineering, Dalian University of Technology, Dalian 116024, *China*

(Received 6 May 2011; revised manuscript received 8 November 2011)

We propose a novel neural network based on a diagonal recurrent neural network and chaos, and its structure and learning algorithm are designed. The multilayer feedforward neural network, diagonal recurrent neural network, and chaotic diagonal recurrent neural network are used to approach the cubic symmetry map. The simulation results show that the approximation capability of the chaotic diagonal recurrent neural network is better than the other two neural networks.

Keywords: diagonal recurrent neural network, chaos, cubic symmetry map

PACS: 87.19.lj, 05.45.Gg **DOI:** 10.1088/1674-1056/21/3/038703

1. Introduction

Neural networks with dynamic characteristics have been a new focus in recent years.[1−8] A dynamic neural network can show the mapping relationship of a dynamic system and deal with the transient stability problem. A diagonal recurrent neural network is a type of dynamic neural network, which shows great improvement as compared with the multilayer feedforward neural network. It can handle the dynamic information by itself for adding an internal feedback channel. Therefore, it is more suitable for modelling and controlling nonlinear systems.[9−14] However, some problems still exist, such as the selection of the number of hidden layer neurons and the limitation of the training algorithm.

In this paper, we construct a novel neural network based on a diagonal recurrent neural network and chaos. Moreover, we design its learning algorithm and use it to approach the cubic symmetry map.

2. Structure of the diagonal recurrent neural network

The diagonal recurrent neural network can realize the dynamic function of the system, and its training speed is quicker than the fully connected recurrent network. The diagonal recurrent neural network can use a gradient descent algorithm to train the parameters of the network because of its characteristics and simple structure.[15]

The structure of typical diagonal recurrent neural network is shown in Fig. 1. There are three layers in the network: input layer, hidden layer, and output layer. The neurons in the hidden layer are dynamic with delayed feedback, which makes the whole network dynamic.

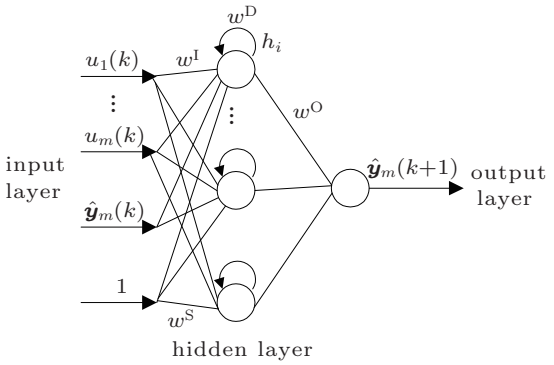


Fig. 1. The structure of the diagonal recurrent neural network.

Let $\boldsymbol{x}(k) = [u_1(k), u_2(k), \cdots, u_m(k), \boldsymbol{y}(k)]^{\mathrm{T}}$ be the input of the network, $s_i(k)$ be the input of the i-th neuron, and $h_i(k)$ be the output of the i-th neuron in the hidden layer. The number of hidden layer neurons is h. The parameters in Fig. 1 are defined as follows: $\boldsymbol{y}(k+1)$ is the output of the network at k+1 time; w_{ij}^{I} is the connection weight of the j-th

*Project supported by the National Natural Science Foundation of China (Grant Nos. 61173183, 60973152, and 60573172), the Superior University Doctor Subject Special Scientific Research Foundation of China (Grant No. 20070141014), and the Natural Science Foundation of Liaoning Province, China (Grant No. 20082165).

†Corresponding author. E-mail: zoey0313@sina.com

© 2012 Chinese Physical Society and IOP Publishing Ltd

neuron in the input layer and the i-th neuron in the hidden layer; w_i^{S} is the threshold of the i-th neuron in the input layer; w_i^{D} is the connection weight of the i-th neuron in the input layer; w_i^{O} is the connection weight of the i-th neuron in the hidden layer and the neuron in the output layer. The input-output relationship of the whole network is as follows:

$$s_i(k) = w_i^{\mathrm{D}} h_i(k-1) + \sum_{j=1}^{m+1} w_{ij}^{\mathrm{I}} x_j(k) + w_i^{\mathrm{S}}, \quad (1)$$

$$h_i(k) = \rho(s_i(k)), \quad (2)$$

$$\boldsymbol{y}(k+1) = \sum_{i=1}^{h} w_i^{\mathrm{O}} h_i(k), \quad (3)$$

where $\rho(\cdot)$ is the activation function of the hidden layer.

3. Learning algorithm of the diagonal recurrent neural network

The diagonal recurrent neural network is trained by the dynamic back-propagation (BP) learning algorithm, then we obtain the function of the error cost $e(k)$ as

$$J(k) = \frac{1}{2}e^2(k) = \frac{1}{2}[d(k) - \boldsymbol{y}(k)]^2,$$

where $d(k)$ is the output of the system.

The adjustment algorithms for the parameters are as follows.

(i) The adjustment algorithm for the connection weight w_i^{O} is as follows:

$$\begin{aligned} w_i^{\mathrm{O}}(k+1) &= w_i^{\mathrm{O}}(k) - \eta \Delta w_i^{\mathrm{O}}(k) \\ &\quad + \alpha(\Delta w_i^{\mathrm{O}}(k) - \Delta w_i^{\mathrm{O}}(k-1)) \\ &= w_i^{\mathrm{O}}(k) - \eta \frac{\partial J(k)}{\partial w_i^{\mathrm{O}}} \\ &\quad + \alpha(\Delta w_i^{\mathrm{O}}(k) - \Delta w_i^{\mathrm{O}}(k-1)) \\ &= w_i^{\mathrm{O}}(k) + \eta e(k) \frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{O}}} \\ &\quad + \alpha(\Delta w_i^{\mathrm{O}}(k) - \Delta w_i^{\mathrm{O}}(k-1)), \quad (4) \end{aligned}$$

$$\frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{O}}} = h_i(k-1), \quad (5)$$

where η is the step, α is the momentum factor, $i = 1, 2, \ldots, h$.

(ii) The adjustment algorithm for the feedback weight w_i^{D} is shown below:

$$\begin{aligned} w_i^{\mathrm{D}}(k+1) &= w_i^{\mathrm{D}}(k) + \eta e(k) \frac{\partial J(k)}{\partial w_i^{\mathrm{D}}} \\ &\quad + \alpha(\Delta w_i^{\mathrm{D}}(k) - \Delta w_i^{\mathrm{D}}(k-1)) \\ &= w_i^{\mathrm{D}}(k) + \eta e(k) \frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{D}}} \\ &\quad + \alpha(\Delta w_i^{\mathrm{D}}(k) - \Delta w_i^{\mathrm{D}}(k-1)), \quad (6) \end{aligned}$$

$$\frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{D}}} = w_i^{\mathrm{O}} \frac{\partial h_i(k-1)}{\partial w_i^{\mathrm{D}}} = w_i^{\mathrm{O}} p_i(k-1), \quad (7)$$

$$\begin{aligned} p_i(k) &= \frac{\partial h_i(k)}{\partial w_i^{\mathrm{D}}} \\ &= \rho'(s_i(k))(h_i(k-1) + w_i^{\mathrm{D}} p_i(k-1)), \quad (8) \end{aligned}$$

where $i = 1, 2, \ldots, h$; $p_i(0) = 0$.

(iii) The adjustment algorithm for the threshold w_i^{S} is

$$\begin{aligned} w_i^{\mathrm{S}}(k+1) &= w_i^{\mathrm{S}}(k) + \eta e(k) \frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{S}}} \\ &\quad + \alpha(\Delta w_i^{\mathrm{S}}(k) - \Delta w_i^{\mathrm{S}}(k-1)), \quad (9) \end{aligned}$$

$$\frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{S}}} = w_i^{\mathrm{O}} \frac{\partial h_i(k-1)}{\partial w_i^{\mathrm{S}}} = w_i^{\mathrm{O}} r_i(k-1), \quad (10)$$

$$\begin{aligned} r_i(k) &= \frac{\partial h_i(k-1)}{\partial w_i^{\mathrm{S}}} \\ &= \rho'(s_i(k))(1 + w_i^{\mathrm{D}} r_i(k-1)), \quad (11) \end{aligned}$$

where $i = 1, 2, \ldots, h$; $r_i(0) = 0$.

(iv) The adjustment algorithm for the connection weight w_{ij}^{I} is

$$\begin{aligned} w_{ij}^{\mathrm{I}}(k+1) &= w_{ij}^{\mathrm{I}}(k) + \eta e(k) \frac{\partial \boldsymbol{y}(k)}{\partial w_{ij}^{\mathrm{I}}} \\ &\quad + \alpha(\Delta w_{ij}^{\mathrm{I}}(k) - \Delta w_{ij}^{\mathrm{I}}(k-1)), \quad (12) \end{aligned}$$

$$\frac{\partial \boldsymbol{y}(k)}{\partial w_{ij}^{\mathrm{I}}} = w_i^{\mathrm{O}} \frac{\partial h_i(k-1)}{\partial w_{ij}^{\mathrm{I}}} = w_i^{\mathrm{O}} q_{ij}(k-1), \quad (13)$$

$$\begin{aligned} q_{ij}(k) &= \frac{\partial h_i(k-1)}{\partial w_{ij}^{\mathrm{I}}} \\ &= \rho'(s_i(k))(x_j(k) + w_i^{\mathrm{D}} q_{ij}(k-1)), \quad (14) \end{aligned}$$

where $i = 1, 2, \ldots, h$; $j = 1, 2, \ldots, m+1$; $q_{ij}(0) = 0$.

4. Structure of the chaotic diagonal recurrent neural network

The structure of the chaotic diagonal recurrent neural network is shown in Fig. 2. There are three layers in the network: input layer, hidden layer, and output layer. The input–output relation of the input layer and the output layer is the same as the general diagonal recurrent neural network. The differences between the chaotic and general diagonal recurrent neural networks lie in the hidden layer. There are two layers in the hidden layer: the first layer is composed of the FB neurons which can receive the forward input and the feedback input, and B neurons which can

only receive the feedback input; the second layer is composed of H neurons, the output of which is really needed. The feedback of FB neurons is the last output of H neurons, and the feedback of B neurons is the last output of B neurons. Each FB neuron connects with all input neurons, and each B neuron connects with only one B neuron. The input of H neurons is the outputs of FB neurons and B neurons, and the output of H neurons is the addition of the outputs of FB neurons and B neurons.

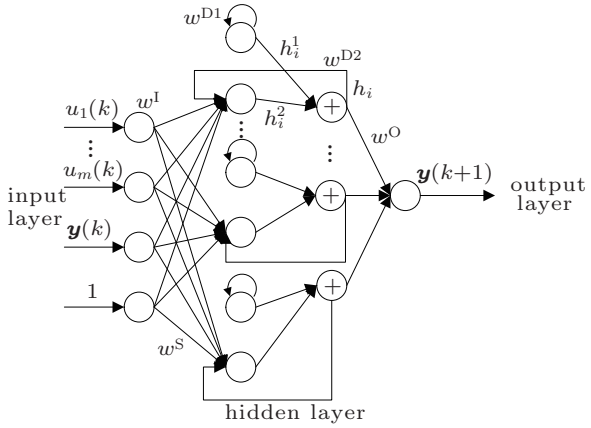


Fig. 2. Structure of the chaotic diagonal recurrent neural network.

Let $\boldsymbol{x}(k) = [u_1(k), u_2(k), \cdots, u_m(k), \boldsymbol{y}(k)]^{\mathrm{T}}$ be the input of the network, $s_i(k)$ be the input of the i-th neuron, and $h_i(k)$ be the output of the i-th neuron in the hidden layer. The number of the hidden layer neurons is h. The parameters in Fig. 2 can be defined as follows: h_i^1 is the output of the i-th B neuron and h_i^2 is the output of the i-th FB neuron; $\boldsymbol{y}(k+1)$ is the output of the network at $k+1$ time; w_{ij}^{I} is the connection weight of the j-th neuron in the input layer and the i-th neuron in the hidden layer; w_i^{S} is the threshold of the i-th neuron in the input layer; w_i^{O} is the connection weight of the i-th neuron in the hidden layer and neuron in the output layer; w_i^{D1} is the connection weight of the i-th B neuron and w_i^{D2} is the connection weight of the i-th FB neuron. The input–output relation of the chaotic diagonal recurrent neural network can be expressed as

$$h_i^1(k) = w_i^{\mathrm{D1}} h_i(k-1)(M - h(k-1)/M, \tag{15}$$

$$h_i^2(k) = \sum_{j=1}^{m+1} w_{ij}^{\mathrm{I}} x_j(k) + w_i^{\mathrm{S}} + w_i^{\mathrm{D2}} h_i(k-1), \tag{16}$$

$$h_i(k) = \rho(h_i^1(k) + h_i^2(k)) = \rho(s_i(k)), \tag{17}$$

$$\boldsymbol{y}(k+1) = \sum_{i=1}^{h} w_i^{\mathrm{O}} h_i(k), \tag{18}$$

where $M > 0$ is a constant, $\rho(\cdot)$ is the activation function of the hidden layer. The activation function of B neurons is the logistic map that can generate chaos.

5. Learning algorithm of chaotic diagonal recurrent neural network

The chaotic diagonal recurrent neural network is also trained by a dynamic BP learning algorithm, and the function of error cost is

$$J(k) = \frac{1}{2}e^2(k) = \frac{1}{2}[d(k) - \boldsymbol{y}(k)]^2.$$

The adjustment algorithms for the parameters are as follows.

(i) The adjustment algorithm for the connection weight w_i^{O} are the same as Eqs. (4) and (5).

(ii) The adjustment algorithm for the feedback weight w_i^{D1} can be written as

$$\begin{aligned} w_i^{\mathrm{D1}}(k+1) &= w_i^{\mathrm{D1}}(k) + \eta e(k)\frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{D1}}} \\ &\quad + \alpha(\Delta w_i^{\mathrm{D1}}(k) - \Delta w_i^{\mathrm{D1}}(k-1)), \end{aligned} \tag{19}$$

$$\frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{D1}}} = w_i^{\mathrm{O}} \frac{\partial h_i(k-1)}{\partial w_i^{\mathrm{D1}}} = w_i^{\mathrm{O}} v_i(k-1), \tag{20}$$

$$\begin{aligned} v_i(k) &= \frac{\partial h_i(k)}{\partial w_i^{\mathrm{D1}}} \\ &= \frac{\partial h_i(k)}{\partial s_i(k)} \left(\frac{\partial h_i^1(k)}{\partial w_i^{\mathrm{D1}}} + w_i^{\mathrm{D2}} v_{i-1}(k-1) \right) \\ &= \frac{\partial h_i(k)}{\partial s_i(k)} (c_i(k) + w_i^{\mathrm{D2}} v_{i-1}(k-1)), \end{aligned} \tag{21}$$

$$\begin{aligned} c_i(k) &= \frac{\partial h_i^1(k)}{\partial w_i^{\mathrm{D1}}} = ((h_i^1(k-1)(M - h_i^1(k-1)) \\ &\quad + w_i^{\mathrm{D1}} c_i(k-1) \\ &\quad \times (M - 2h_i^1(k-1)))/M, \end{aligned} \tag{22}$$

where $i = 1, 2, \ldots, h$; $c_i(0) = 0$.

(iii) The adjustment algorithm for the feedback weight w_i^{D2} can be written as

$$\begin{aligned} w_i^{\mathrm{D2}}(k+1) &= w_i^{\mathrm{D2}}(k) + \eta e(k)\frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{D2}}} \\ &\quad + \alpha(\Delta w_i^{\mathrm{D2}}(k) - \Delta w_i^{\mathrm{D2}}(k-1)), \end{aligned} \tag{23}$$

$$\frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{D2}}} = w_i^{\mathrm{O}} \frac{\partial h_i(k-1)}{\partial w_i^{\mathrm{D2}}} = w_i^{\mathrm{O}} p_i(k-1), \tag{24}$$

$$\begin{aligned} p_i(k) &= \frac{\partial h_i(k)}{\partial w_i^{\mathrm{D2}}} \\ &= \frac{\partial h_i(k)}{\partial s_i(k)} (h_i^2(k-1) + w_i^{\mathrm{D2}} p_i(k-1), \end{aligned} \tag{25}$$

where $i = 1, 2, \ldots, h$; $p_i(0) = 0$.

(iv) The adjustment algorithm for the threshold w_i^{S} is

$$w_i^{\mathrm{S}}(k+1) = w_i^{\mathrm{S}}(k) + \eta e(k)\frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{S}}} + \alpha(\Delta w_i^{\mathrm{S}}(k) - \Delta w_i^{\mathrm{S}}(k-1)), \quad (26)$$

$$\frac{\partial \boldsymbol{y}(k)}{\partial w_i^{\mathrm{S}}} = w_i^{\mathrm{O}}\frac{\partial h_i(k-1)}{\partial w_i^{\mathrm{S}}} = w_i^{\mathrm{O}} r_i(k-1), \quad (27)$$

$$r_i(k) = \frac{\partial h_i(k-1)}{\partial w_i^{\mathrm{S}}} = \frac{\partial h_i(k)}{\partial s_i(k)}(1 + w_i^{\mathrm{D1}} r_i(k-1)), \quad (28)$$

where $i = 1, 2, \ldots, h$; $r_i(0) = 0$.

(v) The adjustment algorithm for the connection weight w_{ij}^{I} is

$$w_{ij}^{\mathrm{I}}(k+1) = w_{ij}^{\mathrm{I}}(k) + \eta e(k)\frac{\partial \boldsymbol{y}(k)}{\partial w_{ij}^{\mathrm{I}}} + \alpha(\Delta w_{ij}^{\mathrm{I}}(k) - \Delta w_{ij}^{\mathrm{I}}(k-1)), \quad (29)$$

$$\frac{\partial \boldsymbol{y}(k)}{\partial w_{ij}^{\mathrm{I}}} = w_i^{\mathrm{O}}\frac{\partial h_i(k-1)}{\partial w_{ij}^{\mathrm{I}}} = w_i^{\mathrm{O}} g_{ij}(k-1), \quad (30)$$

$$g_{ij}(k) = \frac{\partial h_i(k-1)}{\partial w_{ij}^{\mathrm{I}}} = \frac{\partial h_i(k)}{\partial s_i(k)}(x_j(k) + w_i^{\mathrm{D2}} g_{ij}(k-1)), \quad (31)$$

where $i = 1, 2, \ldots, h$; $j = 1, 2, \ldots, m+1$; $g_{ij}(0) = 0$.

6. Simulation results

Three types of neural networks are used to approach the cubic symmetry map, which can be expressed in the following form:[16]

$$f(z) = z(t+1) = \lambda z^3(t) - \lambda z(t) + z(t), \quad (32)$$

where λ is the control parameter, and z is the output of the system. The cubic symmetry map is in chaos when $\lambda \in [3.3, 4]$.

The multilayer feedforward neural network, diagonal recurrent neural network, and chaotic diagonal recurrent neural network are trained 700 times ($\lambda = 3.8$, $z(0) = 0.7$), and the average error is

$$e = \frac{1}{N}\sum_{k=0}^{N}(\boldsymbol{y}(k) - z(k))^2, \quad (33)$$

where $\boldsymbol{y}(k)$ is the output of the neural network.

The simulation results are shown in Fig. 3, where $z(k)$ is shown by the solid lines and $\boldsymbol{y}(k)$ is shown by the dotted lines. To make the figure clearer, the ordinate is magnified 40 times and the abscissa is magnified 5 times. The learning curves of the multilayer feedforward neural network[17] which contains one input neuron, one output neuron, and fifteen hidden layer neurons are shown in Fig. 3(a), and the average error is 0.1456. The learning curves of the diagonal recurrent neural network[18] which is composed of one input neuron, one output neuron, and twenty hidden layer neurons are shown in Fig. 3(b), and the average error is 0.06691. The learning curves of the chaotic diagonal recurrent neural network which is composed of one input neuron, one output neuron, ten FB neurons, ten B neurons, and ten H neurons are shown in Fig. 3(c), and the average error is 0.01018.

When chaotic system (32) is approached by the diagonal recurrent neural network, the average error is 0.06. Because of the dynamic characteristics, the approximation capability of the diagonal recurrent neural network is better than that of the multilayer feedforward neural network. When chaotic system (32) is approached by the chaotic diagonal recurrent neural network, the average error is 0.01. The average errors of multilayer feedforward neural network and diagonal recurrent neural network are 15 times and 7 times higher than that of the chaotic diagonal recurrent neural network, respectively.

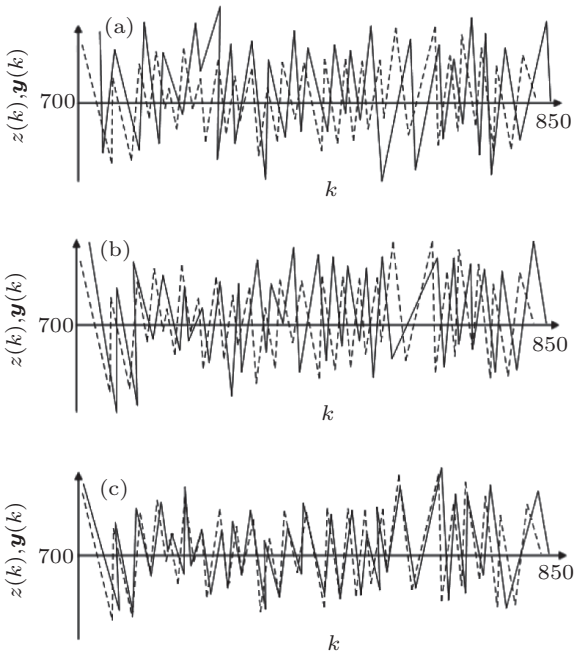


Fig. 3. The learning curves of the neural networks ($\lambda = 3.8$). (a) Multilayer feedforward neural network, (b) diagonal recurrent neural network, (c) chaotic diagonal recurrent neural network.

The three neural networks are also used to ap-

proach the following second-order dynamic system:

$$\begin{cases} \dot{z}_1(k) = z_2(k), \\ \dot{z}_2(k) = -100z_1(k) - 14z_2(k) + 100r(k), \end{cases} \tag{34}$$

with the control signal $r(k) = \sin(2\pi k/20)$, the discrete time nonlinear dynamic system $z(k+1) = z(k)/(1+z^2(k)) + u(k)$, and the training input signal

$$u(k) = \sin\left(\frac{2\pi}{5}k\right) + \sin\left(\frac{2\pi}{15}k\right) + \sin\left(\frac{2\pi}{25}k\right). \tag{35}$$

By doing so, we can obtain the same conclusion.

7. Conclusion

A chaotic diagonal recurrent neural network with a simple structure is proposed, and it is trained by the dynamic BP learning algorithm. Three neural networks are used to approach the chaotic system, the second-order dynamic system, and the discrete-time nonlinear dynamic system. The simulation results show that the approximation capability of the chaos diagonal recurrent neural network is better than the other neural networks.

References

[1] Liu A Q, Shih S C and Wang Q J 2009 *IEEE Trans. Syst. Man Cy.* A **39** 426

[2] Xie W F, Zhu Y Q and Wong Y K 2009 *Neurocomputing* **72** 3277

[3] He G G, Zhu P, Chen H P and Cao Z T 2006 *Acta Phys. Sin.* **55** 1040 (in Chinese)

[4] Lu H T 2002 *Phys. Lett.* A **298** 109

[5] Tang Q and Wang X Y 2010 *Chin. Phys. Lett.* **27** 030508

[6] Zhang H G and Wang Z S 2007 *IEEE Trans. Neural Networ.* **18** 947

[7] Chen S J and Hong Y G 2011 *Chin. Phys.* B **20** 036401

[8] Chen P F, Chen Z Q and Wu W J 2011 *Chin. Phys.* B **20** 040509

[9] Gao F R, Wang F L and Li M Z 2000 *Chem. Eng. Sci.* **55** 1283

[10] Mastorocostas P A and Theocharis J B 2006 *IEEE Trans. Syst. Man Cy.* B **36** 242

[11] Cao J C and Lin X C 2008 *Eng. Appl. Artif. Intel.* **21** 1255

[12] Mastorocostas P A and Hilas C S 2009 *Neural Comput. Appl.* **18** 707

[13] Zhang H G and Luo Y H 2009 *IEEE Trans. Neural Networ.* **20** 1490

[14] Dong M, Zhang H G and Wang Y C 2009 *Neurocomputing* **72** 1999

[15] Ji X D and Familoni B O 1999 *IEEE Trans. Automat. Contr.* **44** 1469

[16] Rogers T D and Whitley D C 1983 *Math. Model.* **4** 9

[17] Aienberg I, Paliy D V, Zurada J M and Astola J T 2008 *Neural Networ.* **19** 883

[18] Gu Z Q and Oyadiji S O 2008 *J. Vib. Acoust.* **130** 061001