Adaptive neural network based stabilization and trajectory tracking control of discrete-time chaotic systems

KURSAD GOKCE[a,*], YILMAZ UYAROĞLU[b]
[a]*Research&Development Center, Otokar Automotive and Defense Ind., Sakarya 54580, Turkey*
[a]*Department of Electrical and Electronics Engineering, Sakarya University, Sakarya 54100, Turkey*

This paper investigates the stability and tracking performance of discrete-time chaotic systems in the presence of external disturbance and noise. For this purpose, a neural network control scheme is developed on the basis of a novel adaptive learning rate to stabilize the chaotic motion of discrete-time chaotic systems to a fixed point as well as to track the desired reference trajectory. The effectiveness of the proposed method is investigated through simulation studies on 2 dimensional Lozi map and performance comparison has been made with well-known backstepping control strategy.

(Received April 22, 2015; accepted June 24, 2015)

Keywords: Adaptive control, Learning systems, Lynapunov, Neural network, Chaos, Nonlinear systems

1. Introduction

Controlling chaos via different control schemes has received great interest in recent years after first introduced by [1]. In OGY–like methods [2-5], when the chaotic trajectory enters a small neighborhood of the corresponding fixed point, a small perturbation is applied to some parameter of the system to keep the chaotic trajectory in the vicinity of the target periodic orbit. However, it is not always an easy task to find a suitable control parameter in the system and a priori knowledge of the unstable periodic orbits is required to apply the control algorithm. Furthermore, these methods are highly sensitive to noise [6].

The control problem of chaotic systems with uncertainty and disturbance has been studied by researchers using different robust control techniques. In [7], a sliding mode controller has been designed to stabilize the unstable periodic orbits of 2 dimensional Hénon map with external disturbances. Sliding mode controllers have been also used to regulate the chaotic systems to their equilibrium state [8-9]. In [10], a fuzzy model based design has been developed for trajectory tracking control of chaotic systems. However, fuzzy models are generally intuitive and require well tuned control parameters.

Backstepping is one of the most promising Lyapunov based adaptive robust control techniques, that has been used for stabilization and tracking control of continuous time [11-13] and discrete time [14] chaotic systems as well as of nonlinear chaotic systems with bounded uncertainties and external disturbances [15-16]. Lynapunov based control techniques has been also applied to synchronization of master-slave Lorenz systems with parameter mismatch [17] and hyperchaotic Yang system with unknown parameters [18].

Recently, neural network (NN) based control algorithms have attracted great interest in control and synchronization of chaotic systems because of their ability to deal with uncertainty and noise. In [19], a NN based algorithm has been utilized to stabilize the chaotic motion in chaotic Hénon map to a desired target trajectory. In [20], the chaotic motions of the Hénon and Logistic maps have been converted to a regular periodic motion by using back propagation NN algorithm. In [21], a similar algorithm has been developed to control the chaotic trajectory of the Ikeda function to equilibrium point.

In this paper, an improved back propagation NN algorithm with adaptive learning rate has been proposed to control the chaotic trajectory of the 2 dimensional discrete chaotic Lozi map to fixed point and to a desired target trajectory. By changing the learning rates adaptively, the control scheme becomes less sensitive to noise and disturbance. In order to evaluate the performance of the proposed method, the discrete-time recursive backstepping control scheme [14] has been also implemented for comparison purpose. The simulation results show that the proposed method with adaptive learning rate outperforms the backstepping control scheme in the presence of noise and disturbance.

2. Discrete time chaotic systems

In this paper, Lozi map is studied as a two dimensional chaotic system. The map equation is given by:

$$\phi(x(k), y(k)) = (x(k+1), y(k+1))$$
$$\phi(x(k), y(k)) = (1 - a|x(k)| + y(k), bx(k)) \quad (1)$$

where $\phi : R^2 \rightarrow R^2$ and it has a chaotic behavior with the parameter $a = 1.7$, and $b = 0.5$ [22] as shown in Fig. 1. In this case, the map in Eq. (1) has 2 fixed points as in Eq. (2) that satisfies $F(x^*) = x^*$.

$$(x_1^*, y_1^*) = \left(\frac{1}{1-b-a}, bx_1^*\right)$$
$$(x_2^*, y_2^*) = \left(\frac{1}{1-b+a}, bx_2^*\right) \quad (2)$$

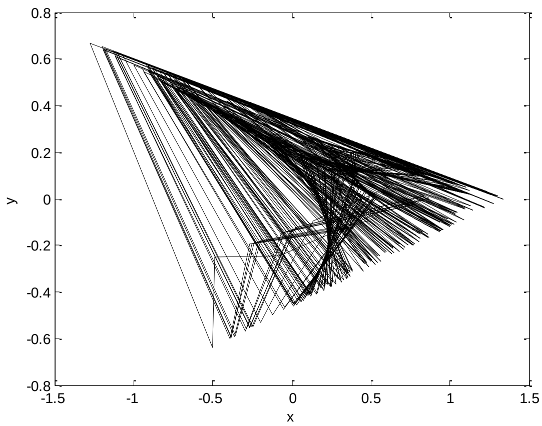

Fig. 1. The attractor of Lozi map.

The goal of this study is to control the chaotic trajectory of Lozi map such that the chaotic trajectory converges to a fixed point or track any desired trajectory even in the presence of some noise and disturbance.

3. Proposed control scheme

The proposed control scheme is a back propagation neural network with adaptive learning rate consisting of 3 layers of neurons (input layer, hidden layer and output layer). The input and output layers have 2 neurons corresponding to the 2 dimensional system given by Eq. (1). The number of hidden neurons affects the learning performance of the network and the determination of the optimal value has been still an open issue in the literature. In this study, it has been selected as 10 to achieve a reasonable performance. The topological structure of the network is shown in Fig. 2.

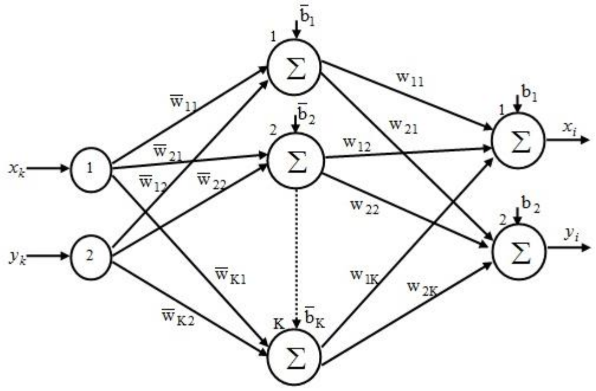

Fig. 2. Neural network topological structure.

Here, (x_k, y_k) $k = 1,2,....N$ are the input patterns generated iteratively from Eq. (1). N is the finite number of input patterns. (x_i, y_i) are the output patterns that should have to be approximated to the fixed points, (x^*, y^*) or to a desired trajectory.

The learning phase of the network has 2 steps. In the first step, the output of the network is calculated based on the network's structure using previous weights and bias values from the first layer to the forward. The following calculations are done in this first step.

The input to kth neuron of the hidden layer is denoted by $\mathrm{I_K}$ and is given by:

$$\mathrm{I_K} = x_k \overline{w}_{K1} + y_k \overline{w}_{K2} + \overline{b}_K \quad (3)$$

where $\overline{w}_{K1}$ and $\overline{w}_{K2}$ are the weights between the kth neuron of the hidden layer. $\overline{b}_K$ is the bias value of the kth neuron in the hidden layer. By applying the tangent sigmoid activation function F on $\mathrm{I_K}$, the output $\overline{O}_{pK}$ is obtained as:

$$F(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

$$\overline{O}_{pK} = F(I_K) \quad (5)$$

The network outputs are given by,

$$x_i = F\left(\sum_{k=1}^{K} \overline{O}_{pK} w_{1K} + b_1\right) \quad (6)$$

$$y_i = F\left(\sum_{k=1}^{K} \overline{O}_{pK} w_{2K} + b_2\right) \quad (7)$$

In the second step, weights and bias values are updated adaptively based on the error values $E_{x_i} = x^* - x_i$,

$E_{y_i} = y^* - y_i$ from the last layer to the back. The adaptation rules for the weight and bias values between the output and hidden layers are calculated by the following equations.

$$w_{1Knew} = w_{1Kold} + \Delta w_{1K}(k+1),\ k = 1,2.....N$$
$$w_{2Knew} = w_{2Kold} + \Delta w_{2K}(k+1),\ k = 1,2....N \quad (8)$$

$$b_{1new} = b_{1old} + \Delta b_1(k+1)$$
$$b_{2new} = b_{2old} + \Delta b_2(k+1) \quad (9)$$

In Eq. (8) and (9), second terms are calculated as:

$$\Delta w_{1K,2K}(k+1) = \eta_{x,y}\delta_{1K,2K}\overline{O}_{pK} \quad (10)$$

$$\Delta b_{1,2}(k+1) = \eta_{x,y}\delta_{1K,2K} \quad (11)$$

$$\delta_{1K} = E_{x_i} x_i(1-x_i)$$
$$\delta_{2K} = E_{y_i} y_i(1-y_i) \quad (12)$$

where η_x and η_y are the adaptive learning rates which is varied according to the error functions and updated as:

$$\eta_{x,y} = \begin{cases} \eta_{x,y}\phi,\ \phi > 1 & \text{if } \left|\mathrm{E}_{x_{i+1},y_{i+1}}\right| \le \left|\mathrm{E}_{x_i,y_i}\right| \\ \eta_{x,y}\varsigma,\ \varsigma < 1 & \text{if } \left|\mathrm{E}_{x_{i+1},y_{i+1}}\right| > \left|\mathrm{E}_{x_i,y_i}\right| \\ \eta_{x,y} & \text{if } \left|\mathrm{E}_{x_i,y_i}\right| \le \mathrm{E}_t \ \text{ for } \forall \mathrm{i} \in \mathrm{N} \end{cases} \quad (13)$$

This rule is applied until a desired error response, E_t, is achieved. In a similar way, the adaptation rules for the weight and bias values between the hidden and input layers are calculated by the following equations:

$$\overline{w}_{K1new} = \overline{w}_{K1old} + \Delta\overline{w}_{K1}(k+1),\ k = 1,2.....N$$
$$w_{K2new} = w_{K2old} + \Delta\overline{w}_{K2}(k+1),\ k = 1,2....N \quad (14)$$

$$b_{Knew} = b_{Kold} + \Delta\overline{b}_K(k+1),\ k = 1,2.....N \quad (15)$$

In Eq. (14) and (15), second terms are calculated as,

$$\Delta\overline{w}_{K1}(k+1) = \eta_z\overline{\delta}_{K1}x_i \quad (16)$$

$$\Delta\overline{w}_{K2}(k+1) = \eta_z\overline{\delta}_{K1}y_i \quad (17)$$

$$\Delta\overline{b}_K(k+1) = \eta_z\overline{\delta}_{K1} \quad (18)$$

$$\overline{\delta}_{K1} = (\delta_{1K}w_{1Kold} + \delta_{2K}w_{2Kold})\,\overline{O}_{pK}\,(1-\overline{O}_{pK}) \quad (19)$$

where η_z is also changed adaptively as:

$$\eta_z = \begin{cases} \eta_z\phi, \phi > 1, \text{if } \left|\mathrm{E}_{x_{i+1}}\right| + \left|\mathrm{E}_{y_{i+1}}\right| \le \left|\mathrm{E}_{x_i}\right| + \left|\mathrm{E}_{y_i}\right| \\ \eta_z\varsigma, \varsigma < 1, \text{if } \left|\mathrm{E}_{x_{i+1}}\right| + \left|\mathrm{E}_{y_{i+1}}\right| > \left|\mathrm{E}_{x_i}\right| + \left|\mathrm{E}_{y_i}\right| \\ \eta_z \text{ , if } \left|\mathrm{E}_{x_i}\right| + \left|E_{y_i}\right| \le \mathrm{E}_t \ \text{ for } \forall \mathrm{i} \in \mathrm{N} \end{cases} \quad (20)$$

With above updating rules, the output of the network follows the desired target trajectory or converges to a fixed point of the map given by Eq. (1).

4. Backstepping control scheme

Backstepping design based on the Lyapunov stability is a recursive procedure that breaks the full system into a sequence of small subsystems. Let the nonlinear system be:

$$\dot{x} = f(x) + g(x)z_1 \quad (21)$$

where $x \in R^n$ is the system state, $z_1 \in R$ is the scalar control input, f and g are nonlinear functions. The system in Eq. (21) can be augmented by the following equations [23]:

$$\dot{z}_1 = f_1(x, z_1) + g_1(x, z_1)z_2$$
$$.$$
$$.$$
$$\dot{z}_k = f_k(x, z_1, ...z_k) + g_k(x, z_1, ...z_k)u \quad (22)$$

where $z_1, z_2, ..z_k \in R$ are the virtual control inputs and $u \in R$ is the final control signal. In backstepping procedure, z_1 is defined as a virtual control input to stabilize the first equation. Then, z_2 is defined as a virtual control input for the second equation, and this goes so on. Therefore, the final control signal, u, for full order system is obtained systematically in n steps [23].

In this study, backstepping control scheme has been applied to the discrete-time chaotic map given by Eq. (1), which can be rearranged as:

$$x_1(k+1) = 1 - a|x_1(k)| + x_2(k)$$
$$x_2(k+1) = bx_1(k) + u(k) \quad (23)$$

The goal is to find $u(k)$ such that the output of the system, $x_1(k)$, asymptotically tracks the reference signal $r(k)$ as well as the stable or unstable fixed points of the system. Backstepping algorithm is applied to the system given by Eq. (23) step by step as follows.

Let $e_1(k) = x_1(k) - r(k)$ be the error between the output of the system and the reference signal. Then:

$$e_1(k+1) = 1 - a|x_1(k)| + x_2(k) - r(k+1) \tag{24}$$

Let the second variable of Eq. (23), $x_2(k)$, be the virtual control input of Eq. (24) and the corresponding error variable $e_2(k) = x_2(k) - \alpha_1(k)$. $\alpha_1(k)$ is the stabilizing function that satisfies the candidate Lyapunov function $V_1(k) = |e_1(k)|$. The derivative of $V_1(k)$ yields:

$$\begin{aligned}\Delta V_1(k) &= |e_1(k+1)| - |e_1(k)| \\ &= |1 - a|x_1(k)| + e_2(k) + \alpha_1(k) - r(k+1)| - |e_1(k)|\end{aligned} \tag{25}$$

If one chooses the stabilizing function as:

$$\alpha_1(k) = a|x_1(k)| - 1 + r(k+1) + c_1 e_1(k) \tag{26}$$

Eq. (25) and (24) becomes:

$$\Delta V_1(k) = |c_1 e_1(k) + e_2(k)| - |e_1(k)| \tag{27}$$

$$e_1(k+1) = c_1 e_1(k) + e_2(k) \tag{28}$$

where c_1 is the design constant to be chosen later. With the stabilizing function given by Eq. (26), the error variable, $e_2(k)$, of the virtual control input is:

$$\begin{aligned}e_2(k+1) &= x_2(k+1) - \alpha_1(k+1) \\ e_2(k+1) &= bx_1(k) + u(k) - c_1(c_1 e_1(k) + e_2(k)) \\ &+ 1 - r(k+2) - a|x_1(k+1)|\end{aligned} \tag{29}$$

Let the candidate Lyapunov function of the full order system based on the error variable of the virtual control input be:

$$V_2(k) = V_1(k) + 2|e_2(k)| \tag{30}$$

If one chooses the control signal, $u(k)$, as:

$$\begin{aligned}u(k) &= a|x_1(k+1)| - 1 + r(k+2) + c_2 e_2(k) \\ &- bx_1(k) + c_1(c_1 e_1(k) + e_2(k))\end{aligned} \tag{31}$$

Eq. (29) becomes:

$$e_2(k+1) = c_2 e_2(k) \tag{32}$$

and the derivative of Eq. (30) becomes:

$$\begin{aligned}\Delta V_2(k) &= \Delta V_1(k) + 2|e_2(k+1)| - 2|e_2(k)| \\ \Delta V_2(k) &\le (|c_1| - 1)|e_1(k)| + (2|c_2| - 1)|e_2(k)|\end{aligned} \tag{33}$$

By choosing design constants appropriately such that $|c_1| < 1,\ |c_2| < 1/2$, Eq. (33) becomes negative definite. Thus, with the control signal given by Eq. (31), $e_1(k) = x_1(k) - r(k) = 0$ as $k \to \infty$ and $x_1(k)$ asymptotically tracks the reference signal $r(k)$. A similar control law can be derived using above procedure in case of $x_2(k)$ is output. If c_1, c_2 are chosen to be zero, deadbeat response is achieved such that $e_1(k)$ goes to zero from any initial state in n (i.e. $n = 2$ for 2 dimensional system) sampling periods [14].

5. Simulation results and discussion

Numerical simulations are carried out to show the effectiveness of the proposed control method. In particular, a comparison with the backstepping control scheme is made to investigate the robustness of the proposed control method in the presence of noise and disturbance.

In this study, 2 control objectives have been considered. The first objective is to stabilize the chaotic trajectory of Lozi map given by Eq. (1) to the fixed point $x_2^* = 0.45,\ y_2^* = 0.22$. The second objective is to track the periodic sinusoidal signal $r(k) = 0.5\sin(k/5)$. In the proposed method, an initial point (0, 0) and its time series have been used to obtain a training set with 500 input patterns. The number of iteration for this training set is 10.

Fig. 3-4 and Fig. 5-6 show the simulation results for the first and second objectives of the proposed method and backstepping control scheme, respectively. The control action has been switched on at $k = 500$.

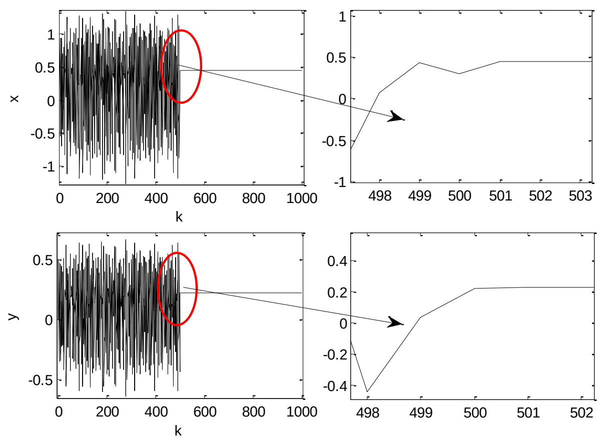

Fig. 3. Time series plot of the system states for the proposed method.

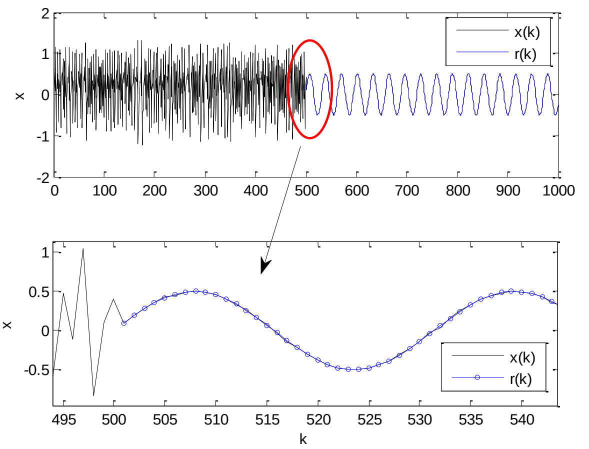

Fig. 4. Tracking performance of the proposed method.

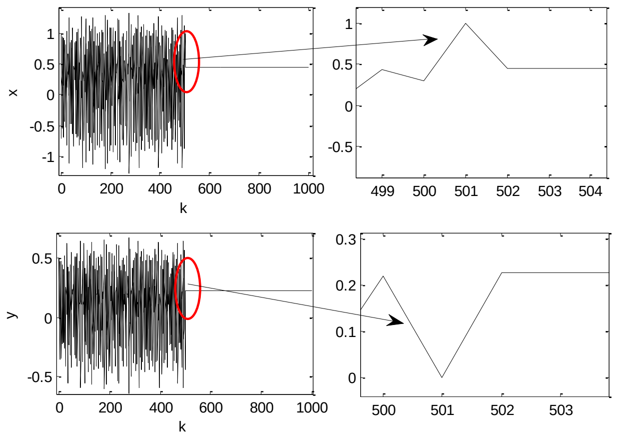

Fig. 5. Time series plot of the system states for the backstepping method.

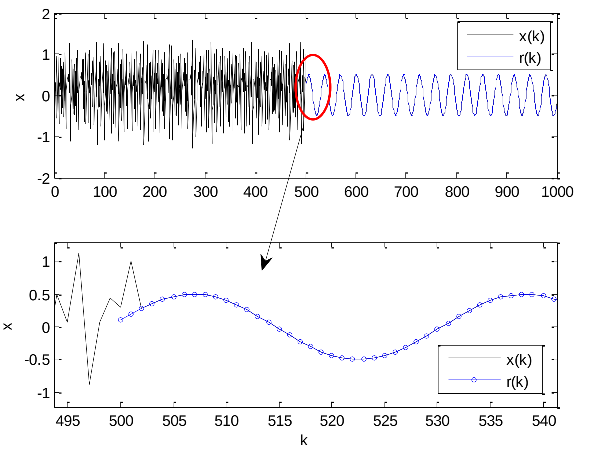

Fig. 6. Tracking performance of the backstepping method.

For backstepping control, design constants have been chosen as $c_1 = c_2 = 0$. As can be seen from Fig. 5, with these constants, deadbeat response has been achieved such that the state variables x and y are stabilized to a fixed point in 2 time steps (i.e. $k = 502$) after control signal is applied.

On the other hand, it can be seen from Fig. 3 that the state variables reach to fixed point after only 1 time step (i.e. $k = 501$) using the proposed control method for this training set.

The number of neurons in hidden layer and the number of iteration in learning process affect the convergence performance of the proposed neural network. The convergence rate of the backstepping control scheme is dependent on the design constants [24]. For $c_1 = 0,\ c_2 = 0$, the quickest convergence is achieved. However, sometimes it is not physically possible to produce a large control signal to achieve deadbeat control, so c_1 and c_2 need to be chosen appropriately. From Fig. 4 and 6, it is obvious that the system state $x(k)$ (i.e. $x_1(k)$) follows the reference signal $r(k)$ successfully using both methods.

In order to show the robustness of the proposed method, a dynamic random noise, distributed on interval (–0.01, 0.01) is added to the state variables. In this case, the tracking performances of both methods have been investigated through Fig. 7.

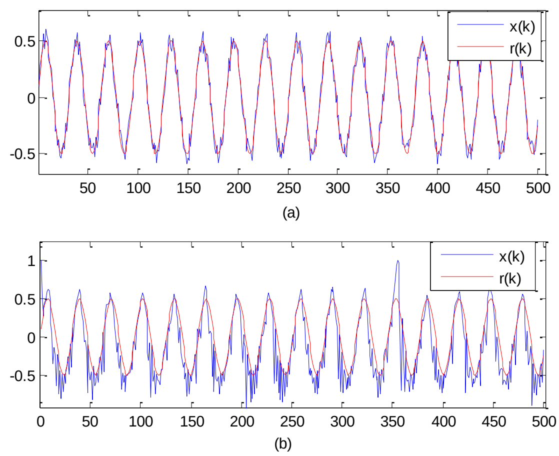

Fig. 7. Tracking performance under noisy measurements (a) proposed method (b) backstepping method.

Fig. 7 shows that tracking error of the proposed method is much smaller than that of backstepping control. Moreover, in order to investigate the performance of the proposed method in the presence of an external disturbance, a disturbance with an amplitude of 0.01 is added to the system states at $k = 800$ after control is switched on at $k = 500$. In this case, the stabilization performances of both methods have been investigated through Fig. 8.

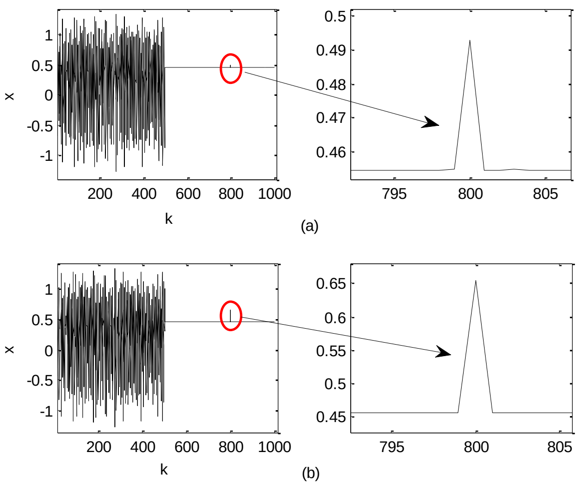

Fig. 7. Stabilization performance under an external disturbance applied (a) proposed method (b) backstepping method.

Here, it can be observed that the proposed method is more robust with respect to noise and disturbance than backstepping control if the training parameters are chosen properly.

5. Conclusions

In this paper, a neural network control based on the adaptive learning rates has been proposed to control the chaotic trajectories of the discrete-time chaotic systems. In order to compare the performance of the proposed method, a backstepping control scheme has been also developed and applied to the chaotic system. Simulation studies demonstrated that the proposed method can successfully stabilize the chaotic trajectory of the system to the fixed point and track the desired reference trajectory. It has been also shown that the proposed method is less sensitive to noise and disturbance and has a more robust property than backstepping control scheme if the neural network is trained properly with accurate training parameters. The proposed method can be applied to several circuits and systems studied in the literature and be used for both stabilization and tracking problems of chaotic systems.

References

[1] E. Ott, C. Grebogi, J. A. Yorke, Phys. Rev. Lett., **64**, 1196 (1990).
[2] J. D. Skufca, E. M. Bollt, Physica D, **179**, 18 (2003).
[3] O. J. Kwon, J. D. Park, H. Lee, J. Korean Phys. Soc., **38**, 635 (2001).
[4] T. K. Dutta, A. K. Jain, International Journal of Statistika and Mathematika, **7**, 67 (2013).
[5] H. H. Tsai, C. C. Fuh, P. C. Tung, J. Mar. Sci Technol., **10**, 77 (2002).
[6] Z. Hong, Y. Jie, W. Jiao, W. Yinghai, Phys. Rev. E, **53**, 299 (1996).
[7] G. Cai, W. Zhou, Z. Tan, International Journal of Nonlinear Science, **4**, 118 (2007).
[8] U. E. Kocamaz, Y. Uyaroğlu, H. Kizmaz, Int. J. Adapt. Control Signal Process., **28**, 1413 (2014).
[9] Y. Uyaroğlu, M. Varan, S. Emiroğlu, Optoelectron. Adv. Mater. – Rapid Comm. **6**, 498 (2012).
[10] K. Y. Lian, J. J. Liou, Proc. of the 16th IFAC World Congress, 1119 (2005).
[11] Q. Zhang, Y. Shu, W. Zhang, Journal of East China Normal University, **1**, 52 (2010).
[12] S. Mascolo, G. Grassi, Int. J. Bifurcat. Chaos, **9**, 1425 (1999).
[13] O. S. Onma, O. I. Olusola, A. N. Njah, J. Nonlinear Dynamics, **2014**, 1 (2014).
[14] J. Lu, R. Wei, X. Wang, Z. Wang, IEEE T Circuits-I, **48** (1359), 2001.
[15] Y. C. Chang, Asian Journal of Control, **15**, 1752 (2013).
[16] R. Luo, Int. J. Adapt. Control Signal Process, (2015).
[17] L. Liu, H. Song, X. Feng, J. Hu, G. Wang, Optoelectron. Adv. Mater. – Rapid Comm. **7**, 154 (2013).
[18] H. Zhu, C. Zeng, Optoelectron. Adv. Mater. – Rapid Comm. 8, 100 (2014).
[19] L. D. Mei, W. Zheng, Commun. Theor. Phys., **40**, 439 (2003).
[20] X. P. Zong, Y. Geng, Int. Conference on Wavelet Analysis and Pattern Recognition, 166 (2009).
[21] C. Hernandez, A. Martinez, J. Castellanos, F. L. Mingo, The 6th IEEE International Conference on Electronics, Circuits and Systems, 1231 (1999).
[22] J. C. Sprott, Oxford University Press: Chaos and Time-Series Analysis, 148 (2003).
[23] M. Krstic, I. Kanellakopoulos, P. Kokotovic, Wiley: Nonlinear and Adaptive Control Design, 1995.
[24] T. Wu, M. S. Chen, Physica D, **164**, 53 (2002).

*Corresponding author: kgokce@otokar.com.tr