Nonlinear Analysis: Modelling and Control, 2005, Vol. 10, No. 2, 95–106

Neural Predictive Control of Unknown Chaotic Systems

A. Boukabou[1], **N. Mansouri**[2]

[1]Department of Electronics, Jijel University, BP 98, Jijel 18000, Algeria

[2]Department of Electronics, Mentouri University, Constantine 25000, Algeria
aboukabou@mail.univ-jijel.dz

Received: 20.02.2005
Accepted: 11.04.2005

Abstract. In this work, a neural networks is developed for modelling and controlling a chaotic system based on measured input-output data pairs. In the chaos modelling phase, a neural network is trained on the unknown system. Then, a predictive control mechanism has been implemented with the neural networks to reach the close neighborhood of the chosen unstable fixed point embedded in the chaotic systems. Effectiveness of the proposed method for both modelling and prediction-based control on the chaotic logistic equation and Hénon map has been demonstrated.

Keywords: chaos, neural networks, predictive control.

1 Introduction

Due to the structural complexity of the chaotic systems and due to their extreme sensitivity to initials conditions, the problem of controlling or ordering such systems has received increasing attention in recent years. Specially, E. Ott, C. Grebogi and J. Yorke [1] introduced a control method of chaotic systems known as OGY method, according to which the unstable fixed point can be stabilized by applying only small variations to a control parameter. Since that, significant attention has been focused on developing techniques for the control of chaotic systems [2–9].

In the efforts of identifying and controlling chaos, there are difficulties in modelling chaotic systems represented only by the input-output data pairs provided by the underlying system. Moreover, there is no effective method to design

controllers to ensure good tracking performance. Therefore, it is necessary to develop an approach that solve such problems.

Recently, neural networks have found success as a promising way to solve highly nonlinear control problem [10–13]. Application of neural networks for chaos control have been successfully applied [14–17]. In [16], neural networks have been used for controlling chaotic dynamical systems trained in the OGY method and the Pyragas method. In these techniques, emphasis is placed on control of a chaotic system to a desired target. Ren *et al.* [17] proposed a dynamic neural network control method for unknown continuous nonlinear systems. The control structure includes a dynamic neural network identifier and a model-based neural network controller.

In this paper, we are looking into the possibility of developing a simple neural network model for both modelling and controlling discrete chaotic systems given only by input-output data pairs. This is done by combining the neural networks for modelling and the predictive controller is directly designed based on the online network model for the control of the unknown chaotic system on one of its unstable fixed point. Moreover, we give very simple necessary and sufficient conditions for local stabilization of unstable fixed points by the proposed method.

The paper is organized as follows. In Section 2, we introduce the neural networks modelling incorporating with feed-forward backpropagation functions and a Levenberg-Marquardt computational algorithm is used for calculation. Based on this modelling approach, a neural predictive controller is developed and detailed in Section 3. The chaotic logistic equation and the Hénon system are then used as examples to illustrate the advantages and control performance of the proposed approach in Section 4. Finally, conclusion is given in Section 5.

2 Neural network

For a neural networks with $\big(x(i), y(i)\big)$ input output data pairs, $i = 1, 2, \ldots, n$ number of iterations and one hidden layer of $j = 1, 2, \ldots, m$ neurons.

For a such network, the input to j^{th} units of the hidden layer denoted by I_j is the weighed sum of all the inputs added to the bias

$$I_j = \sum_{i=1}^{n} w_{ji} x(i) + b_{ji}, \tag{1}$$

where w_{ji} are the interconnection weights from the jth function and the ith input to the output and the b bias. The input I_j is then passed through an activation function to produce an output O_j as shown in Fig. 1.

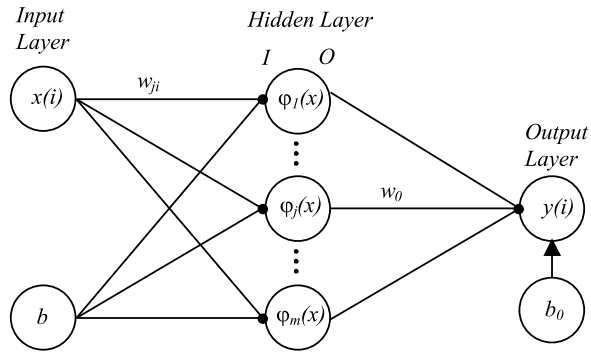

Fig. 1. Training of the neural network for modelling and controlling phases.

The activation function is represented with a function F. The output O_j is given by

$$O_j = F(I_j) = F\Big(\sum_{j=1}^{m} w_{ji}x(i) + b_j\Big), \tag{2}$$

The network output is given by

$$y(i) = F\big(x(i)\big) = \sum_{j=1}^{m} w_o\varphi_j\big(x(i)\big) + b_o. \tag{3}$$

There are many ways to define the activation function. One that is often used in neural networks is the *sigmoid function* which is used in this network

$$\varphi_j\big(x(i)\big) = \frac{1}{1 + \exp\big(-b_{ji} - w_{ji}x(i)\big)}. \tag{4}$$

The neural network has to be trained such that it can perform both modelling and controlling tasks. Initially, the network weights w_{ji} and bias b_{ji} of the hidden layer and the weight w_o and bias b_o of the output layer are assigned randomly. The output $y(i)$ is calculated for each associated input $x(i)$. In order to minimize

the error between the network output $y(i)$ and the output of the true system $y_s(i)$, weights and bias are adjusted to minimize the error vector $e_{\max}$ given by

$$e_{\max} = \frac{1}{2}\sum_{i=1}^{n} e_i^2 = \frac{1}{2}\sum_{i=1}^{n} \big(y_s(i) - y(i)\big)^2. \tag{5}$$

Note that $e_{\max}$ is a function of w_{ji}, b_{ji}, w_o and b_o. We can use the Levenberg-Marquardt method on $e_{\max}$ to update w_{ji}, b_{ji}, w_o and b_o. The amount of update for each parameter is

$$\alpha_{new} = \alpha_{old} - (J_\alpha^T.J_\alpha + \lambda.I)^{-1} J_\alpha^T e_i, \tag{6}$$

where α is a vector whose elements are the different parameters of the neural model, I is the identity matrix, the quantity $\lambda > 0$ is called the *momentum term* which will help to keep the updates moving in the right direction and J_α represents Jacobian for each parameters defined by

$$J_{w_{ji}} = \frac{\partial\varphi\big(x(i)\big)}{\partial w_{ji}},\ J_{b_{ji}} = \frac{\partial\varphi\big(x(i)\big)}{\partial b_{ji}},\ J_{w_o} = \frac{\partial F\big(x(i)\big)}{\partial w_o},\ J_{b_o} = \frac{\partial F\big(x(i)\big)}{\partial b_o}. \tag{7}$$

The Levenberg-Marquardt algorithm could be run many reprises to try to train the neural network system to much data pairs very well. The algorithm runs until all of the parameters stop moving or change very little over a series of update steps. This indicates that the value of the error is minimized, so the algorithm has found a minimum and it can be terminated. However, parameters are reinitialized if the error is over the desired tolerance or if the maximum number of iteration is reached.

3 Neural predictive controller design

In this section, the control objective is to suppress chaos, that is, to drive the unknown chaotic system from a chaotic regime to a regular attractor such as fixed points.

Fig. 2 shows the block diagram of the neural predictive control where the plant under control is the unknown chaotic system, y_s is the output of the true system, y is the neural output, e_i is the modelling error, y_p is the predicted output of the neural system and u is the control.

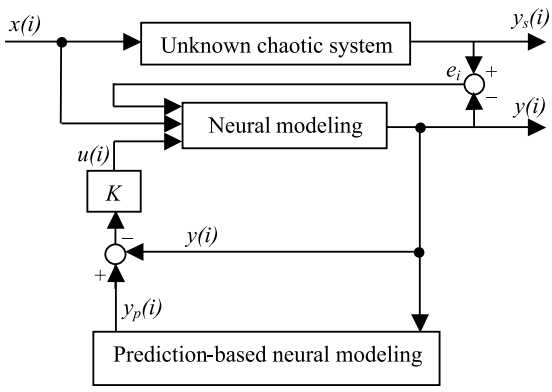

Fig. 2. The block diagram.

Suppose that the neural model of the unknown chaotic system has an unstable fixed point $\bar{y}$, it satisfy:

$$\bar{y} = F(\bar{x}) \tag{8}$$

and is currently in a chaotic state. The purpose of predictive control is to assure the system asymptotically converges towards $\bar{y}$with only extremely small applied force u. In order to avoid such behavior, we design a conventional feedback controller $u(i)$ added to the dynamical system (3) of the form

$$y(i) = F\big(x(i)\big) + u(i), \tag{9}$$

$u(i) \in \Re^n$ is determined by the difference between the predicted states and the current states. It is chosen in such a way to make the trajectory of the uncontrolled system converge to an unstable fixed point $\bar{y}$.

$$u(i) = K\big(y_p(i) - y(i)\big), \tag{10}$$

where K is an adjustable coefficient of the controller, $y_p(i)$ is the predicted future state of uncontrolled chaotic systems from the current state $y(i)$.

Based on the neural model established above, neural prediction of a future state y_p of uncontrolled unknown chaotic system is determined by

$$y_p(i) = y(i+p) = F\big(x(i+p)\big). \tag{11}$$

Using a one step ahead-prediction, the controlled unknown chaotic system is then given by:

$$y(i) = F\big(x(i)\big) + K\big(F\big(x(i+1)\big) - F\big(x(i)\big)\big). \tag{12}$$

The simplest way to formulate an applicable control law is to make use of the fact that the dynamics of any smooth nonlinear system is approximately linear in a small neighborhood of a fixed point. Thus, near $\bar{y}$, we can use the linear approximation for the uncontrolled system by

$$\delta y(i) = A\delta F\big(x(i)\big), \tag{13}$$

where

$$\delta y(i) = y(i) - \bar{y} \quad \text{and} \quad \delta F\big(x(i)\big) = F\big(x(i)\big) - \bar{x} \tag{14}$$

and A is the derivatives evaluated at the fixed point $\bar{y} = F(\bar{x})$.

The controlled system is linearized around $\bar{y}$ by

$$\begin{aligned}
\delta y(i) &= A\delta F\big(x(i)\big) + \delta u(i) \\
&= A\delta F\big(x(i)\big) + K\big(\delta y(i+1) - \delta y(i)\big) \\
&= A\delta F\big(x(i)\big) + K\big(\delta F\big(x(i+1)\big) - \delta F\big(x(i)\big)\big) \\
&= A\delta F\big(x(i)\big) + K\big(A\delta F\big(x(i)\big) - \delta F\big(x(i)\big)\big) \\
&= \big(A + K(A-1)\big)\delta F\big(x(i)\big).
\end{aligned} \tag{15}$$

The control problem is to design $u(i)$ to control the system state y to track the unstable fixed point $\bar{y}$. Gain K is computed such that equation (15) is exponentially stable. This implies that the controller gain must satisfy the following inequality:

$$\big|A + K(A-1)\big| < 1. \tag{16}$$

Furthermore, determining unstable fixed points experimentally needs that at least one input-output data pair traveled on, which is not necessary satisfied, in this case, we assume that if

$$\big|y(i) - y(i-1)\big| < \varepsilon \tag{17}$$

for ε a small positive number. Then the neural model is in the vicinity of the fixed points.

Since there is no mathematical model of the chaotic system, values of A can be determined by simulation in the vicinity of the unstable fixed point by

$$A = \frac{y(i+1) - y(i)}{y(i) - y(i-1)}. \tag{18}$$

Similar to the OGY method [1], in order to apply the proposed predictive control strategy, we have to determine the correction to apply in the vicinity of the fixed point to adjust the next point so it falls on the fixed one. Stability is guaranteed in a neighborhood of the stabilized fixed point, and the controlled neural system become as follow

$$y(i) = \begin{cases} F\big(x(i)\big) + u(i), & \text{if} \quad \big|y(i) - y(i-1)\big| < \varepsilon, \\ F\big(x(i)\big), & \text{otherwise.} \end{cases} \tag{19}$$

All things considered, we are now able to apply the proposed approach and test its efficiency for both modelling and controlling unknown chaotic systems which will be the objective of the next section.

4 Simulation study

4.1 Example: logistic equation

In this example, the logistic mapping is defined by a quadratic function of the form

$$y_s(i) = p y_s(i-1)\big(1 - y_s(i-1)\big). \tag{20}$$

When $p = 3.75$, the system is chaotic without control.

The system can be rewritten by:

$$\begin{aligned} x(i) &= y_s(i-1), \\ y_s(i) &= p x(i)\big(1 - x(i)\big), \end{aligned} \tag{21}$$

where $\big(x(i), y_s(i)\big)$ are the measured input and output of the unknown chaotic systems.

We first randomly generated 1000 data pairs using the true system, and then used them to train the neural model.The neural network is characterized by one inputs x, a hidden layer of 10 neurons $(m = 10)$ and one output y. Levenberg-Marquardt algorithm is used for best search of new parameters w_{ji} and the bias b_{ji} of the hidden layer, the weight w_o and b_o of the output layer.

The plot of input output data pairs generated from the unknown chaotic system, the resulting neural model compared to the plot of the true system and the modelling error are shown in Figs. 3(a), (b). The output data is plotted with "+" marker and the corresponding neural model outputs is plotted with solid line. They are indistinguishable.

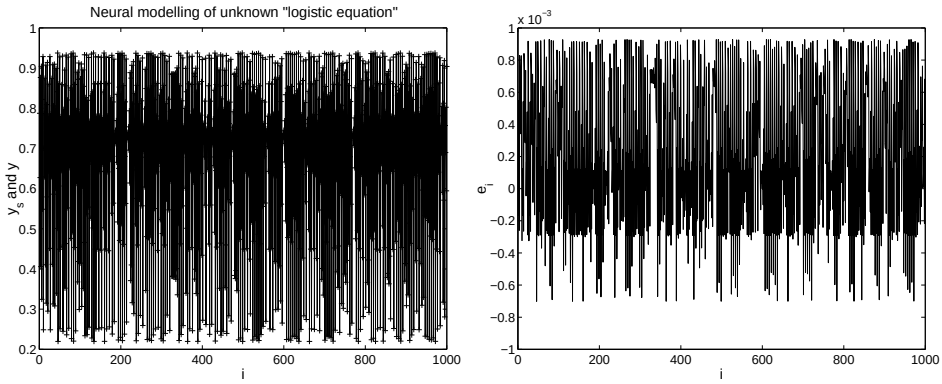

Fig. 3. Measured output data and neural model (a); modelling error (b).

Once the neural model of the unknown logistic equation is established and in order to apply the proposed neural predictive control strategy, we have to determine the correction to be applied in the vicinity of the fixed point to adjust the next point so it falls on the fixed one. We have to compute A and gain K.

We start simulation with $y(0) = 0.7$ as initial condition. At time $i = 216$, the test is verified and we get $y(216) = 0.7294$.

From equation (18), value of A is obtained from simulation as follow:

$$A = \frac{y(i+1) - y(i)}{y(i) - y(i-1)} = \frac{0.7401 - 0.7294}{0.7294 - 0.7356} = -1.7258. \tag{22}$$

The controlled system around its fixed point is given by

$$\delta y(i) = (-1.7258 - 2.7258K)\delta F\big(x(i)\big). \tag{23}$$

The fixed point is stabilized by the proposed method of control if

$$|-1.7258 - 2.7258K| < 1. \tag{24}$$

Thus, the feedback gain K is in the range of:

$$-1 < K < -0.2663 \tag{25}$$

we choose $K = -0.65$.

In the validation phase of the identified neural model, once inequality (17) satisfied, control input u switch on and tracks the trajectory towards the unstable fixed point. Simulation results in Figs. 4(a), (b) show that control of the neural system at the fixed point works very well.

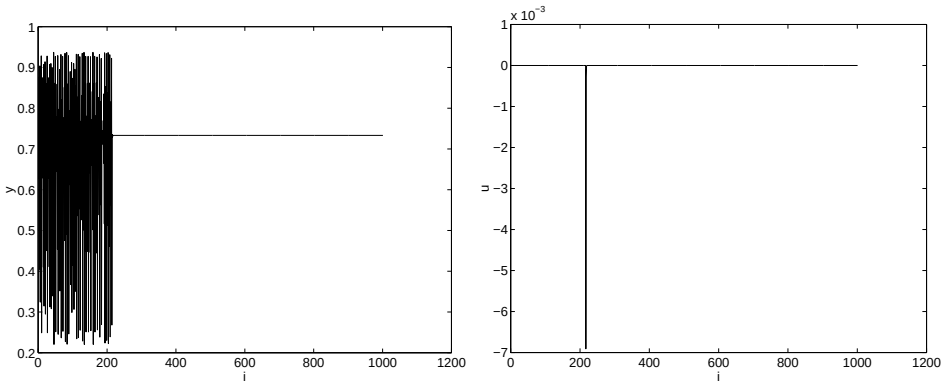

Fig. 4. Neural model under predictive control (a); control input (b).

At $i = 217$, appropriate states is close to the unstable fixed point, the control input takes a nonzero value and stabilize trajectory on unstable fixed point.

4.2 Example: Hénon system

Simulation on some higher-dimensional systems have been also carried out. Experience shows that, just like many other methods, the higher the dimension is, the longer the simulation time will be. However, in terms of quality of control performance, they are the same. We show a two-dimensional example. The Hénon system is a nonlinear system with chaotic behavior for certain values of

parameters, which is represented by the relation:

$$y_s(i) = a - y_s(i-1)^2 + by_s(i-2). \tag{26}$$

For $a = 1.4$ and $b = 0.3$, system exhibit a chaotic regime. It can be rewritten by:

$$\begin{aligned} x_1(i) &= y_s(i-1), \\ x_2(i) &= y_s(i-2), \\ y_s(i) &= a - x_1(i)^2 + bx_2(i). \end{aligned} \tag{27}$$

In order to control the unknown chaotic system given only by input-output data pairs, the system is first modelized using the neural network. In the modelling phase, neural network is characterized by two inputs x_1 and x_2, a hidden layers of 30 neurons ($m = 30$) and one output y.

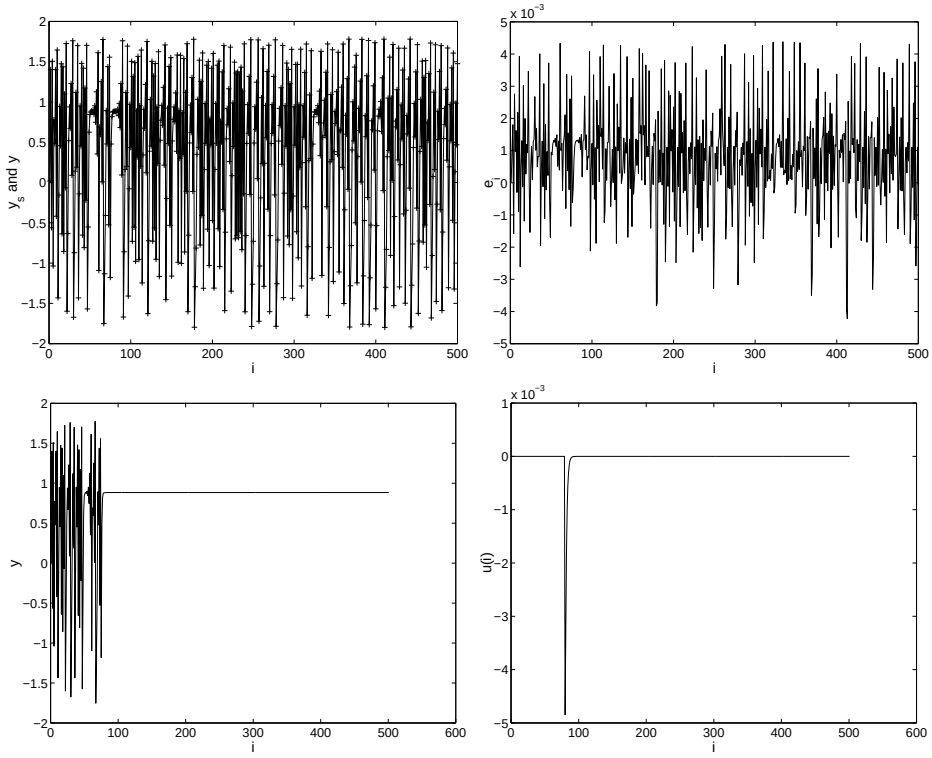

Fig. 5. Measured output data and neural model (a); modelling error (b); neural model under predictive control (c); control input (d).

We start simulation with $x(0) = 0.01$ and $x(1) = 0.02$ as initial condition. At time $i = 80$, $y(80) = 0.8856$. A is obtained from simulation as follow

$$A = \frac{y(i+1) - y(i)}{y(i) - y(i-1)} = \frac{0.8800 - 0.8856}{0.8856 - 0.8813} = -1.3023. \tag{28}$$

The controlled system around its fixed point is given by

$$\delta y(i) = (-1.3023 - 2.3023K)\delta F\big(x(i)\big). \tag{29}$$

The fixed point is stabilized by the proposed method of control if

$$|-1.3023 - 2.3023K| < 1. \tag{30}$$

This yield to $-1 < K < -0.1313$ we choose $K = -0.8$.

The simulation results of neural modelling and neural controlling phases of the unknown chaotic Hénon system are summarized altogether in Fig. 5.

5 Conclusion

In this paper, a neural predictive control method has been completely designed, analyzed and successfully applied to the control of unknown discrete chaotic systems, where the neural model is established using only the input-output data pairs of the underlying system. Based on this neural model, the proposed control law ensures that the chaotic state tracks stable constant targets, which falls on fixed points. The performance of the proposed control was demonstrated using logistic equation and Hénon system. We conclude that the suggested schemes can effectively solve the control problems of unknown chaotic systems based on neural models.

References

1. E. Ott, C. Grebogi, J.A. Yorke. Controlling chaos, *Phys. Rev. Lett.*, **64**, pp. 1196–1199, 1990.

2. W.L. Ditto, S.N. Rauseo, M.L. Spano. Experimental control of chaos, *Phys. Rev. Lett.*, **65**, pp. 3211–3214, 1990.

3. E.R. Hunt. Stabilizing high-period orbits in a chaotic system: the diode resonator, *Phys. Rev. Lett.*, **67**, pp. 1953–1955, 1991.

4. K. Pyragas. Continues control of chaos by self-controlling feedback, *Phys. Lett. A*, **170**, pp. 421–428, 1992.

5. G. Chen, X. Dong. *From Chaos to Order Methodologies, Perspectives and Applications*, World Scientific, Singapore, 1998.

6. T. Ushio, S. Yamamoto. Prediction-based control of chaos, *Phys. Lett. A*, **264**, pp. 30–35, 1999.

7. A.M. Harb, M.A. Zohdy. Chaos and bifurcation control using nonlinear recursive controller, *Nonlinear Analysis: Modelling and Control*, **7**(2), pp. 37–43, 2002.

8. C. Piccardi, S. Rinaldi. Control of complex peak-to-peak dynamics, *Int. J. of Bifurcation and Chaos*, **12**(12), pp. 2927–2936, 2002.

9. A. Boukabou, N. Mansouri. Controlling chaos in higher-order dynamical systems, *Int. J. of Bifurcation and Chaos*, **14**(11), pp. 4019–4025, 2004.

10. H. Nguyen, B. Windrow. Neural networks for self-learning control systems, *IEEE Control System Mag.*, **18**, pp. 18–23, 1990.

11. W.T. Miller, R.S. Sutton, P.J. Werbos. *Neural Networks for Control*, The MIT Press, Cambridge, MA, 1991.

12. D.T. Pham, X. Liu. *Neural Networks for Identification, Prediction and Control*, Springer-Verlag, UK, 1995.

13. A.S. Poznyak, W. Yu, E.N. Sanchez, J.P. Perez. Nonlinear adaptive trajectory tracking using dynamic neural networks, *IEEE Trans Neural Networks*, **10**(11), pp. 1402–1411, 1999.

14. M. Alsing, A. Gavrielides, V. Kovanis. Using neural networks for controlling chaos, *Phys. Rev. E*, **49**, pp. 1225–1231, 1994.

15. K.B. Kim, J.B. Park, Y.H. Choi, G. Chen. Control of chaotic dynamical systems using radial basis function network approximators, *Information Sciences*, **130**, pp. 165–183, 2000.

16. M. Ramesh, S. Narayanan. Chaos control of Bonhoeffer-van der Pol oscillator using neural networks, *Chaos Solutions and Fractals*, **12**, pp. 2395–2405, 2001.

17. X.M. Ren, A.B. Rad, P.T. Chan, W.L. Lo. Identification and control of continuous-time nonlinear systems via dynamic neural networks, *IEEE Trans Industrial Electronics*, **50**(3), pp. 478–486, 2003.