Control of chaos by linear and nonlinear feedback methods

斉藤, 孝則
九州大学大学院総合理工学府

伊藤, 早苗
九州大学応用力学研究所

矢木, 雅敏
九州大学応用力学研究所

https://doi.org/10.15017/3550

出版情報：九州大学応用力学研究所所報. 127, pp.27-38, 2004-09. Research Institute for Applied Mechanics, Kyushu University
バージョン：
権利関係：

Control of chaos by linear and nonlinear feedback methods

Takanori SAITO *1, Sanae-I. ITOH *2, Masatoshi YAGI*2

E-mail of corresponding author: *tsaito@riam.kyushu-u.ac.jp*

(Received July 30, 2004)

Abstract

The chaotic orbits in Rossler system is controlled into a periodic cycle by two methods; the delayed feedback controlling method which continues to control chaos by self-controlling feedback, and a control system including on-line trained neural network controller. It is found that (1) the stabilization of chaotic orbit by former method depends on the choice's of initial conditions and gain parameters and that (2) the linear neural controller fails to control chaotic orbit in Rossler system and the choice of the threshold function (nonlinear function) is found to be essential in the second methods.

Key words : *chaos, delayed feedback controlling method, neural controller*

1. Introduction

The chaotic phenomena are frequently observed in plasma discharges and in the fundamental plasma experiments[1]. The recent research extends not only to identify the chaos but also to control the chaos itself[2]. In fusion plasmas, the control of plasma turbulence is critical issue to attain the self-ignition conditions. Some dynamics of ELM (Edge Localized Model) in fusion plasma can be explained by using the system with a few degrees of freedom. In addition, inside of internal transport barrier(ITB)[1] in a toroidal plasma, the turbulence is suppressed and transport reduces to the level of neoclassical values. In such a situation, a model with a few degrees of freedom is still applicable. The technique of controlling chaos gives the fundamental concept and can be applied to that for fusion plasmas. Generally, chaos is defined by a positive Lyapunov exponent. Plasma turbulence is on one hand categorized as a hyper chaos and might be characterized by several positive Lyapunov exponents. To establish the controlling method of plasma turbulence, we investigate the controlling of simple chaos as a first step. The original controlling method of chaos is called OGY method developed by Ott, Grebogi and Yorke[3]. A chaotic orbit is stabilized by applying small perturbations into the system. However, this method requires the knowledge of the location of unstable fixed point (UFP) in the system so that the location should be tracked in advance. Later, Pyragas proposed the delayed feedback controlling method[4], which does not require the knowledge of the location of UFP. The alternative method of controlling chaos is to apply a Neural Network Controller (NNC). Recently, the NNC has been proposed by Konishi and Kokame which is applied to control the chaotic orbit in the two dimensional map system[5,10]. This method does not require knowledge of the location of the desired UFP or UPO.

In this thesis, both methods (delayed feedback controlling method and neural controller) are tested to Rossler system. In chapter 2, we review OGY method, delayed feedback controlling method, neural controller and also a genetic algorithm as an alternative method for advancing the weights in neural controller. In chapter 3, the control of chaos system using delayed feedback controlling method is studied. In chapter 4, the control of chaos system using a neural controller is investigated. The advantage and disadvantage of this method are discussed comparing with the delayed feedback method. In chapter 5, summary and discussion are given. Application of genetic algorithm for NNC is briefly discussed in Appendix.

2. Review

2.1 OGY method

The pioneers of controlling chaos are Ott, Grebogi, and Yorke[3]. They have proposed so called OGY method, which utilizes the existence of UFP embedded in the chaotic attractor(See Fig.1). The OGY method is explained as follows: If we first determine some of

*1 Interdisciplinary Graduate School of Engineering Sciences, Kyushu University

*2 Research Institute for Applied Mechanics, Kyushu University

the unstable low-period periodic orbits embedded in the chaotic attractor, then we examine these orbits and choose one which yeilds improved system performance. Finally, we tailor our small time-dependent parameter perturbations so as to stabilize this already exsiting orbit(See Fig.2). This method requires the knowledge of the location of UFP so that the location should be tracked in advance by the linear prediction.

The map system is generally written as

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), u(k)), \tag{1}$$

where $\mathbf{x}(k)$ is the state vector, $u(k)$ is the control signal. A fixed point $\mathbf{x}_f$ of the map system is defined by

$$\mathbf{x}_f = f(\mathbf{x}_f, 0). \tag{2}$$

Then, the linearized map system at the fixed point is given by

$$\mathbf{x}(k+1) - \mathbf{x}_f = A(\mathbf{x}(k) - \mathbf{x}_f) + \mathbf{b}u(k), \tag{3}$$

where $A = \frac{df(\mathbf{x}_f,0)}{dx}$, $\mathbf{b} = \frac{df(\mathbf{x}_f,0)}{du}$. We set λ_u, λ_s as eigenvalues of 2×2 matrix A and $\mathbf{e}_u$, $\mathbf{e}_s$ as the corresponding eigenvectors. We introduce 2×2 matrix P which consists of the first column $\mathbf{e}_u$ and the second column $\mathbf{e}_s$

$$p = (\mathbf{e}_u, \mathbf{e}_s). \tag{4}$$

Inverse matrix P^{-1} is given by

$$p^{-1} = \begin{pmatrix} \mathbf{v}_u \\ \mathbf{v}_s \end{pmatrix}, \tag{5}$$

where $\mathbf{v}_u$, $\mathbf{v}_s$ are raw vectors, respectively. Using the relation $P^{-1}P = I$, we obtain the following relations

$$\mathbf{e}_s\mathbf{v}_u = \mathbf{e}_u\mathbf{v}_s = 0 \quad \mathbf{e}_u\mathbf{v}_u = \mathbf{e}_s\mathbf{v}_s = 1 \tag{6}$$

and

$$A = p\begin{pmatrix} \lambda_u & 0 \\ 0 & \lambda_s \end{pmatrix}p^{-1} = \lambda_u\mathbf{e}_u\mathbf{v}_u + \lambda_s\mathbf{e}_s\mathbf{v}_s \tag{7}$$

If $\mathbf{x}(k+1)$ is in the neighbor of $\mathbf{x}_f$, then $\mathbf{x}(k+1) - \mathbf{x}_f$ is parallel to $\mathbf{e}_s$. Therefore, the relation holds:

$$\mathbf{v}_u(\mathbf{x}(k+1) - \mathbf{x}_f) = 0. \tag{8}$$

From eqs. (2.3), (2.7) and (2.8) , we obtain

$$\lambda_u(\mathbf{x}(k) - \mathbf{x}_f) + \mathbf{v}_u\mathbf{b}u(k) = 0. \tag{9}$$

Finally, $u(k)$ is given by

$$u(k) = \begin{cases} -\frac{\lambda_u}{\mathbf{v_u b}}(\mathbf{x}(k) - \mathbf{x}_f) & \|\mathbf{x}(k) - \mathbf{x}_f\| < \epsilon \\ 0 & \end{cases}, \tag{10}$$

where ϵ is a small positive value.

Now let us consider the Henon map:

$$\begin{cases} X_1(k+1) = 1.4 - X_1(k)^2 + 0.3X_2(k) + U(k), \\ X_2(k+1) = 0.3X_1(k) \end{cases} \tag{11}$$

Linearing (2.11) at the fixed point X_f, we obtain

$$A = \begin{pmatrix} -2\mathrm{x}_f & 0.3 \\ 1 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{12}$$

Hence the eigenvalues and eigenvectors of A are given by

$$\lambda_u = -X_f - \sqrt{X_f^2 + 0.3}, \quad \lambda_s = -X_f + \sqrt{X_f^2 + 0.3} \tag{13}$$

$$e_u = \begin{pmatrix} \frac{\lambda_u}{\sqrt{\lambda_u^2+1}} \\ \frac{1}{\sqrt{\lambda_u^2+1}} \end{pmatrix}, \quad e_s = \begin{pmatrix} \frac{\lambda_s}{\sqrt{\lambda_s^2+1}} \\ \frac{1}{\sqrt{\lambda_s^2+1}} \end{pmatrix}, \tag{14}$$

$$v_u = \begin{pmatrix} \frac{\sqrt{\lambda_u^2+1}}{\lambda_u - \lambda_s} & -\frac{\lambda_s\sqrt{\lambda_u^2+1}}{\lambda_u - \lambda_s} \end{pmatrix},$$
$$v_s = \begin{pmatrix} \frac{\sqrt{\lambda_s^2+1}}{\lambda_s - \lambda_u} & -\frac{\lambda_u\sqrt{\lambda_s^2+1}}{\lambda_s - \lambda_u} \end{pmatrix}. \tag{15}$$

Using these relations, $u(k)$ is written as

$$u(k) = -\lambda_u \begin{pmatrix} 1, & -\lambda_s \end{pmatrix} (\mathbf{x}(k+1) - \mathbf{x}_f). \tag{16}$$

Fig.3 shows the iterations of X in the Henon map. By OGY method, the system can be stabilized to the period-one cycle as shown in Fig.4. The control parameters $X_f = 0.8, \epsilon = 0.1$ are used, and is applied during the iterations k=1000-3000. The OGY method is generally inadequate when the system is far from the fixed point, and the method is also limited to the case for controlling only one or two-dimensional maps.

2.2 Delayed feedback controlling method

Pyragas proposed the delayed feedback controlling method by modifying the OGY method, which does not require the knowledge of the location of the desired UFP or UPO. The method is based on the idea of the stabilization of UFPs or UPOs embedded in the chaotic attractor. This is achieved by making a time-dependent perturbation into the system. There are the infinite number of different UPOs within the chaotic attractor. Due to the infinite number of different UPOs in strange attractor, chaotic system can be tuned to a large number of distinct periodic regimes. This is done by switching the temporal programing of small parameter perturbation to stabilize different periodic orbits. The method can stabilize only those periodic orbits whose maximal Lyapunov exponents is small compared to the reciprocal of the time interval between parameter changes. Since the corrections of the parameter are rare and small, the fluctuation noise leads to occasional bursts of the system into the region far from the desired periodic orbit and these bursts are more frequent for large noise[3]. In addition, the changes of parameter are discrete in time, since OGY method deals with the Poincare map. On the other hand, the delayed feedback controlling method controls chaos by self-controlling delayed feedback, continuously not discretely and is not sensitive to the noise.

This method is useful to maintain the control of practical chaotic systems, which are influenced by a change in their environment (e.g., temperature, humidity, atmospheric pressure and so on.)

It is assumed that the equations to describe the system are unknown, however a certain scalar variable could be measured as a system output, and then the system has an input available for delayed output signal $y(t-\tau)$. These assumptions can be met by the following model:

$$\begin{cases} \dfrac{dy}{dt} = P(y, \mathbf{x}) + F(t) \\ \dfrac{d\mathbf{x}}{dt} = \mathbf{Q}(y, \mathbf{x}) \\ F(t) = K(y(t-\tau) - y(t)) = KD(t) \end{cases} \tag{17}$$

Here τ is a delayed time, K is the gain of perturbation, and the vector $\mathbf{x}$ describes the remaining variables of the dynamic system which are not available or not of interest for observation. The delayed time τ corresponds to the period of the UPO, since the relation $y(t) = y(t-\tau)$ is hold for $D(t) = 0$. τ is chosen numerically such that it gives the local minimum value of $D^2(t)$. K can be regarded as a criterion of UPO stabilization. When this stabilization is achieved, K reduces to an extremely small value. As well as in the OGY method, a small perturbation is used to stabilize the UPOs. Therefore choosing an appropriate weight value of K of the feedback, one can achieve the stabilization. The delayed time τ and the weight value of K of the feedback should be adjusted in an experiment. By this method, an experimental realization is very simple. If appropriate values of K for systems are found, this method is applicable to a variety of systems.

2.3 Neural controller

Neural controller proposed by Konishi and Kokame consists of a watcher and Neural Network Controller (NNC). Fig.5 shows the block diagram of a neural controller. Here the NNC consists of two layers such as input layer and output layer. Let us consider the following chaotic system:

$$\mathbf{X}(t+1) = \mathbf{F}\{\mathbf{X}(t)\} + \mathbf{B}\mathbf{U}(t), \tag{18}$$

where $\mathbf{X}(t)$ is the state vector, $\mathbf{F}$ denotes chaotic system and $\mathbf{U}(t)$ is the control signal. A fixed point $\mathbf{X}_f$ of the map system is given by

$$\mathbf{X}_f = \mathbf{F}(\mathbf{X}_f). \tag{19}$$

The local dynamics of chaotic system at the fixed point is approximately governed by

$$\xi(t+1) = \mathbf{H}\xi(t) + \mathbf{B}\mathbf{U}(t), \qquad \xi(t) = \mathbf{X}(t) - \mathbf{X}_f. \tag{20}$$

The matrix $\mathbf{H}$ is the local linearized map of $\mathbf{F}$ at $\mathbf{X}_f$.

$$\mathbf{H} = \frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}} \tag{21}$$

When the orbits $\mathbf{X}(t)$ and $\mathbf{X}(t-1)$ of the chaotic system without control satisfy

$$\|\mathbf{X}(t) - \mathbf{X}(t-1)\| < \epsilon, \tag{22}$$

the watcher passes a control signal from NNC to the chaotic system, and then the weights of connection from the input neuron to output neuron in NNC are updated by back propagation method. An initial value of the weight is determined by a random number. ϵ is assumed to be a small positive value. Fig.6 shows that NNC consists of two input neurons and two output neurons. The NNC is governed by

$$U_i(t_k) = f(O_i(t_k) + \theta_i(k)), \tag{23}$$

$$O_i(t_k) = \sum_{j=1}^{2} W_{ij}(k) X_j(t_k), \tag{24}$$

where j is the number of input neurons, i, the number of output neurons and k is the number of times at which the NNC is trained(i.e., the number of times the watcher operates), W_{ij} is the weight of output neuron, and $O_i(t_k)$ is the signal which goes into neuron of output layer through the weight from ith neuron of input layer, and $\theta_i(k)$ is the bias of the ith neuron of the output layer. The weights and the biases are updated by using

$$W_{ij}(k+1) = W_{ij}(k) + \Delta W_{ij}(k), \tag{25}$$

$$\theta_i(k+1) = \theta_i(k) + \Delta\theta_i(k), \tag{26}$$

where

$$\Delta W_{ij}(k) = -\eta \frac{\partial E(k)}{\partial U_i(n_k)} \frac{\partial U_i(n_k)}{\partial W_{ij}(k)}, \tag{27}$$

$$\Delta\theta_i(k) = -\eta \frac{\partial E(k)}{\partial U_i(n_k)} \frac{\partial U_i(n_k)}{\partial \theta_i(k)}, \tag{28}$$

η is the learning rate and the error is defined by

$$E(k) = k_C E_C(k) + k_U E_U(k), \tag{29}$$

with

$$E_C = \frac{1}{2}[\mathbf{X}(t_k+1) - \mathbf{X}(t_k)]^T [\mathbf{X}(t_k+1) - \mathbf{X}(t_k)], \tag{30}$$

and

$$E_U = \frac{1}{2}\mathbf{B}\mathbf{U}(t_k)^T \mathbf{B}\mathbf{U}(t_k), \tag{31}$$

where $\mathbf{B}$ is a known constant matrix. Also, k_C and k_U are the error weights for $E_C(k)$ and $E_U(k)$, respectively. $E_C(k)$ is an error function corresponding to the Euclidean norm of $[\mathbf{X}(t_k+1) - \mathbf{X}(t_k)]$, and $E_U(k)$ is an error function corresponding to the Euclidean norm of the input signal $[\mathbf{B}U(t_k)]$. The partial derivative $\partial E(k)/\partial U_i(t_k)$ in eqs(2.27),(2.28) is given by

$$\frac{\partial E(k)}{\partial U_i(t_k)} = k_c \frac{\partial E_c(k)}{\partial U_i(t_k)} + k_u \frac{\partial E_u(k)}{\partial U_i(t_k)} \tag{32}$$

$$= k_c \sum_{j=1}^{N} b_{ij}(X_j(t_k+1)-X_j(t_k))+k_u \sum_{j=1}^{N} b_{jl} \sum_{l=1}^{m} b_{jl}U_l(t_k). \tag{33}$$

After the NNC has been sufficiently trained, the control signal $U(t_k)$ of the trained NNC can be described by

$$U(t_k) = -\frac{1}{1+\zeta}\mathbf{B}^{-1}[\mathbf{H}-\mathbf{I}][\mathbf{X}(t_k)-\mathbf{X}_f], \tag{34}$$

$$\begin{cases} \mathbf{X}(t_k) \in \delta, \\ \zeta = \frac{k_u}{k_c} \in (-1,+\infty) \end{cases} \tag{35}$$

Equation (2.34) produces the control signal $U(t_k)$ such that the controlled orbit $X(t_k+1)$ is located in a line between $\mathbf{H}[\mathbf{X}(t_k)-\mathbf{X}_f]+\mathbf{X}_f$ and $\mathbf{X}(t_k)$ (see Fig.7). The position of the controlled orbit $\mathbf{X}(t_k+1)$ in the line depends on the ratio of the error weights, ζ. If the NNC has been trained by a proper ratio ζ, the chaotic orbit is stabilized onto the UFP.

Fig.8 shows Henon map without control. The bold dot embedded in the chaotic attractor indicates a desired UFP. The Henon map with a neural network controller is given by

$$\begin{cases} X_1(t+1) = -0.1X_1(t) + X_2(t) + U_1(t) \\ X_2(t+1) = X_1(t)^2 - 1.6 + U_2(t) \end{cases} \tag{36}$$

The control system is used as $\eta = 0.01$, $\zeta = 1.0$ ($k_c = k_u = 1.0$). Figure 2.9 shows the result of the stabilization of the chaotic system. It follows from Fig.9 that the stabilization of the map is successfully achieved without a knowledge of the location of the UFP.

2.4 Genetic algorithm

A Genetic Algorithm (GA) is a method for searching solutions to a given problem rapidly and efficiently. The fitness of given solution can be determined; a solution with a high fitness value is better than a solution with a low fitness value, although the maximum possible fitness might not be known. GA performs well in cases for which gradient information is not available; in such situations, an algorithm such as the conjugate-gradient method cannot be used to find maxima in the solution space. The basic idea of GA is to consider an ensemble, or population, of possible solutions to the problem. A genetic algorithm consists of a description of solutions, a fitness function, and methods (such as crossover and mutation) as shown in Fig.10. First, a population of randomly generated solutions is formed and the fitness of each solution is determined. After the population has been evaluated, the best solutions are copied(1). These copies are changed slightly, with the hope that some of these random changes will produce a better solution; this process is called mutation(2). Additionally, randomly chosen elements from pairs of good solutions are combined to form new solutions; this process is called crossover(3). A new population is assembled from the mutated copies, the new solutions are formed from crossover, and the best solutions are formed from the original population. The fitness evaluation of the population and the creation of new solutions is repeated until an adequate solution is found. Each evaluation of the population is a generation. In this manner, the process is evolving.

A genetic algorithm is used to evolve neurons which are used to form neural networks for stabilization of an unstable fixed point[6]. This method is a replacement of backpropagation which requires gradient information. The advantage of the genetic algorithm approach for the chaos control problem is that the network weights can be found by examining the performance of a network as a controller rather than by providing correct control signals for various input data. Fig.11 shows the schematic view of NNC with GA. Each output layer neuron is specified by its weights W_{ij}. A network is formed with several neurons selected from the population. The fitness of a network is used to determine the fitness of the individual neurons. The fitness function is a statement of the goals of GA. The proper choice of the fitness function determines the speed with which the GA can converge on the correct solution.

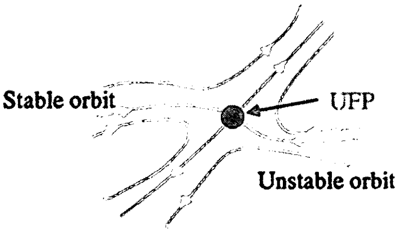

Fig. 1 UFP embedded in chaotic attractor

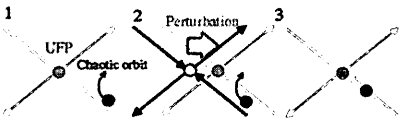

Fig. 2 OGY method

3. Control of chaos system using delayed feedback controlling method

In this chapter, we applied a delayed feedback controlling method to Rossler system to control chaos. Rossler system with the perturbation $F(t)$ is written by

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + 0.2 + F(t) \\ \frac{dz}{dt} = 0.2 + z(x-5.7) \end{cases} \tag{37}$$

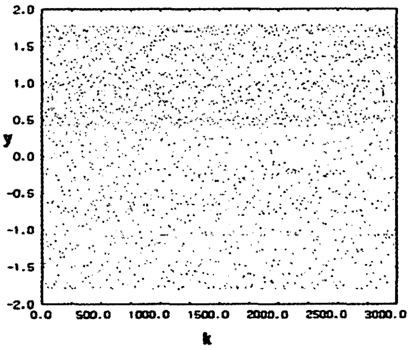

Fig. 3 Dependence of y on k

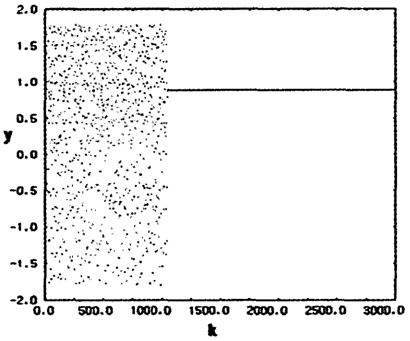

Fig. 4 Dependence of y on k with OGY method

with

$$\begin{cases} F(t) = K[y(t-\tau) - y(t)] = K[D(t)] \quad (t > 50) \\ F(t) = 0 \quad (t \leq 50) \end{cases}$$

where K is a gain of perturbation and τ is the delayed time explained in chapter 2. Fig.12 shows chaotic attractor of Rossler system without control. Fig.13 shows the $x-y$ phase portrait of attractor. Fig.14 shows the dependence of the time averaged perturbation $< D^2(t) >$ on τ. The time average is performed during the time interval $t = 150 - 300$. $< D^2(t) >$ is calculated for the prescribed value of τ with different initial conditions such as $x_0 = y_0 = z_0 = 0.5, 1.0, 1.5, \cdots, 10$. τ is scanned from 0 to 20 with an interval of 0.01. The corresponding twenty values of $< D^2(t) >$ for each τ are shown. The value of K which gives local minimum values of $< D^2(t) >$ for a fixed τ is also found. Fig.15

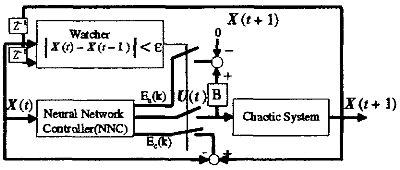

Fig. 5 Block diagram of a neural controller

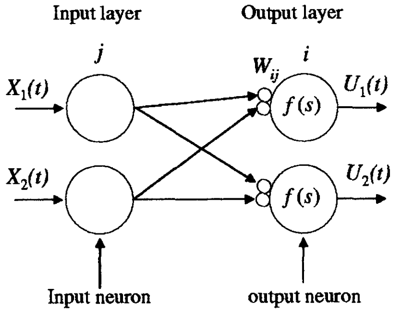

Fig. 6 Neural network controller

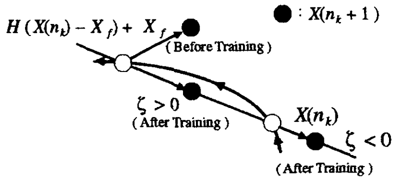

Fig. 7 Schematic diagram of the controlled orbit

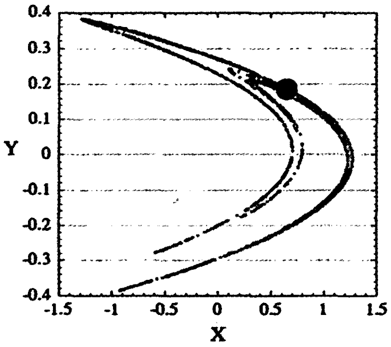

Fig. 8 Henon map

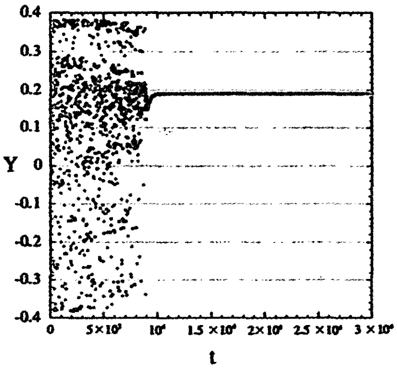

Fig. 9 Stabilization of chaos orbit in Henon map

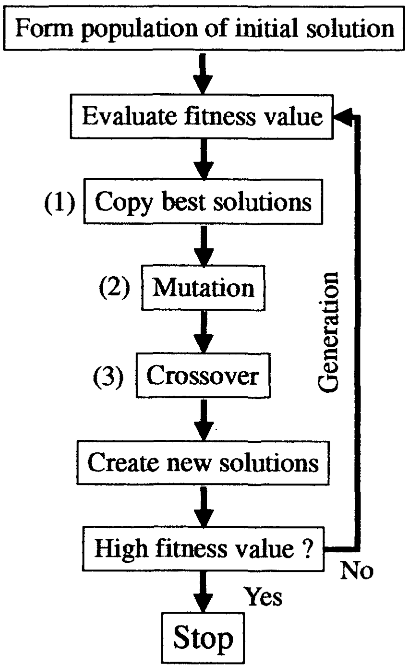

Fig. 10 Flow chart of Genetic algorithm

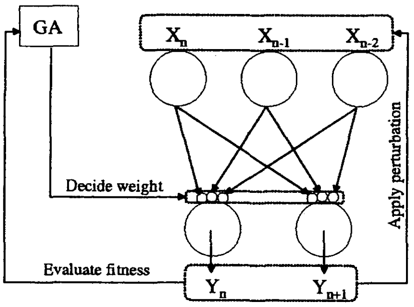

Fig. 11 Block of diagram Genetic algorithm

shows the dependence of $< D^2(t) >$ on K in the cases with $\tau = 5.9$, $\tau = 11.75$ and $\tau = 17.5$, respectively. It is seen that $K = 0.18$, $K = 0.12$ and $K = 0.06$ correspond to the local minimal of $< D^2(t) >$ for period one ,two and three cycles, respectively. Fig.16 shows an example of failure for controlling chaos in the case with $K = 0.8$, $\tau = 5.9$ for the initial condition $x_0 = y_0 = z_0 = 5.5$. The chaotic attractor is seen. Fig.17 shows an example of a success for the control in the case with $K = 0.18$, $\tau = 5.9$, the same initial condition as those in Fig.16 The period-one cycle is seen as the result of stabilization. Fig.18 shows the time evolution of y and F in this case. The origin of the curve F corresponds to the time when the perturbation is switch on. The perturbation becomes small after a transient process and the system comes into the periodic regime, which initially corresponds to an unstable orbit. As well as the period-one cycle of the Rossler system, a period-two cycle can be stabilized by choosing an appropriate weight K of the perturbation. Fig.19 shows an example of a success for the control of period-two orbit in the case with the initial condition $x_0 = y_0 = z_0 = 5.5$. It is found that the control of chaos on the period-one cycle ($\tau = 5.9$) or period two cycle ($\tau = 11.75$) is insensitive or, does not depend on the initial condition. On the other hand, it depends on the initial condition for the control of period-three cycle ($\tau = 17.5$). Fig.20 shows an example of a success for the control of period-three orbit in the case with $x_0 = y_0 = z_0 = 5.5$. Fig.21 shows an example of failure in the case with $x_0 = y_0 = z_0 = 9.5$. In any cases, the choice of K is found to be important to stabilize UPO. It is concluded that this method is sensitive to the choice of values of τ, K and initial values. In addition, the chaotic orbits of Rossler system are found to converge to UFPs in the cases, which correspond to $\tau = 3.3$ and $\tau = 9.3$. In these cases, the value of K weakly affects the stabilization of UFP compared with the stabilization of UPO.

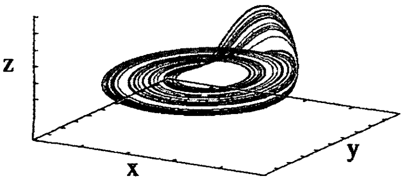

Fig. 12 Chaotic attractor of Rossler system

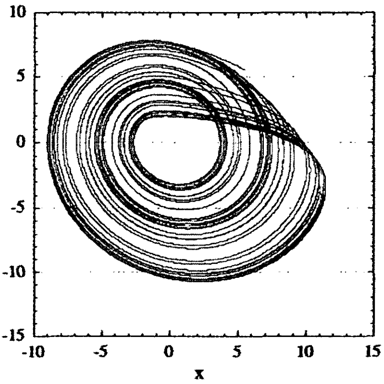

Fig. 13 x-y phase portrait of Rossler system

4. Control of chaos system using a neural controller

In this chapter, we apply a neural network controlling method to Rossler system. Fig.22 shows the block

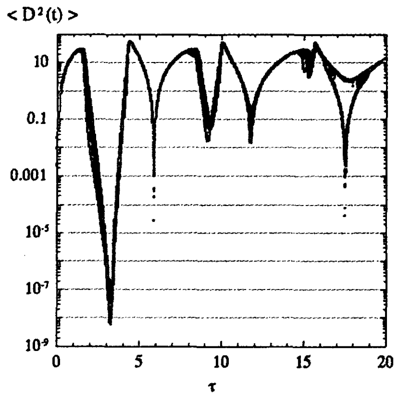

Fig. 14 Dependence of $< D^2(t) >$ on τ

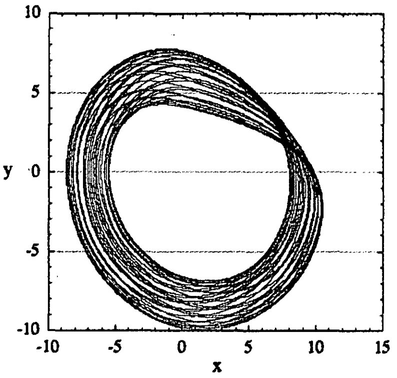

Fig. 16 x-y phase portrait of the Rossler system($\tau = 5.9, K = 0.8, t > 100$)

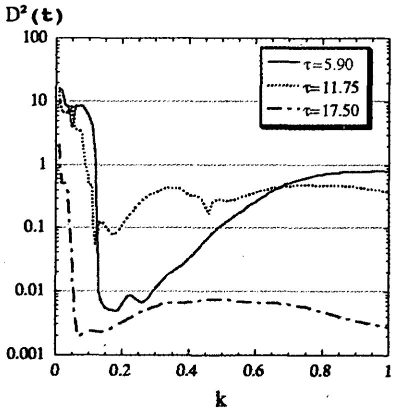

Fig. 15 Dependence of $D^2(t)$ on K

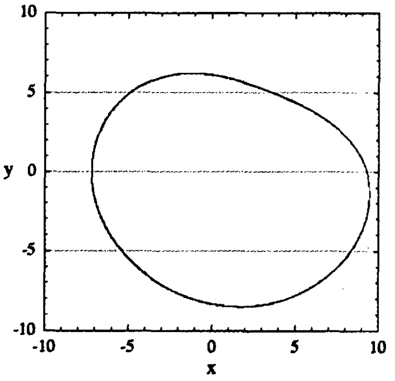

Fig. 17 x-y phase portrait of the Rossler system($\tau = 5.9, K = 0.18, t > 100$)

diagram of the original neural controller. When the orbits $X(t)$ and $X(t-\tau)$ of the chaotic system without control satisfy

$$\|\mathbf{X}(t) - \mathbf{X}(t-\tau)\| < \epsilon \tag{38}$$

the watcher passes a control signal from NNC to the chaotic system. We apply a neural controller to stabilize UPO in Rossler system with three-dimensional chaotic system in the following

$$\left\{ \begin{array}{l} \frac{dx}{dt} = -y - z + U_1(t) \\ \frac{dy}{dt} = x + 0.2 + U_2(t) \\ \frac{dz}{dt} = 0.2 + z(x - 5.7) + U_3(t) \end{array} \right. \tag{39}$$

In this case, NNC consists of three input neurons and three output neurons as shown in Fig.23. In the following, we examine the performance of threshold functions and the sensitivity to the initial conditions. Three cases with different threshold functions are investigated for period-one, two and three: (1) linear function $f(x) = x$, (2) the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$, (3) the hyperbolic tangent function $f(x) = \frac{1-e^{-x}}{1+e^{-x}}$. Parameters : $\epsilon = 1.0$, $\eta = 0.08$, $k_C = 3.0$ and $k_U = 1.0$ are used. The definitions are made in equations (2.22),(2.27),(2.28) and (2.29). It is found that the case (1) fails in the application to all period cases, the case (2) fails in the application to period-two and three cases for $k_U = 1.0$, however, if we change $k_U = 4.2$, then it could control period-three. The case (3) works for all cases without changing a parameter k_U. Two cases with different initial values on the period-three cycle ($\tau = 17.5$) are investigated $x = y = z = 5.5$, $x = y = z = 9.5$. It is found that the control of chaos does not depend on the choice of the initial condition. These results are shown in Fig.24, Fig.25 and Fig.26 for period-one, two and three cases, respectively. We analized power spectrum to find the periodic orbits of the chaotic system. Fig.27 shows the dependence of the power spectrum on the frequency with/without control for $\tau = 5.9$. The peak is clearly observed at $f = 0.17$ for the case with control, which implies to correspond to period-one cycle ($f = 1/\tau$). Two cases with $\tau = 11.75$ and $\tau = 17.5$ are shown in Fig.28. Corresponding peaks are observed at

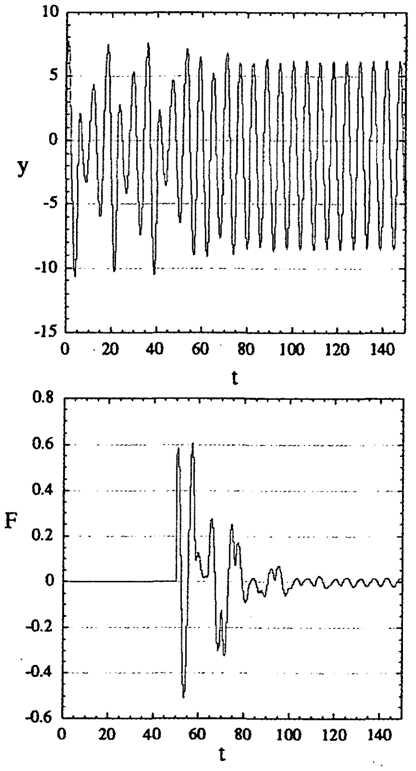

Fig. 18 Stabilization of period-one cycle of Rossler system

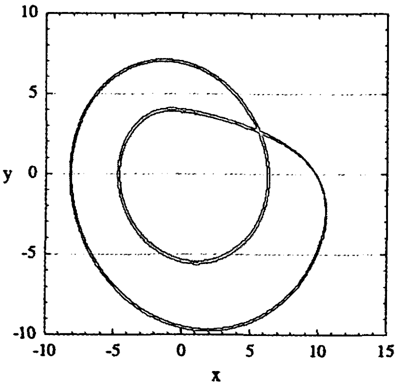

Fig. 19 x-y phase portrait of the Rossler system($\tau = 11.75, K = 0.12, t > 100$)

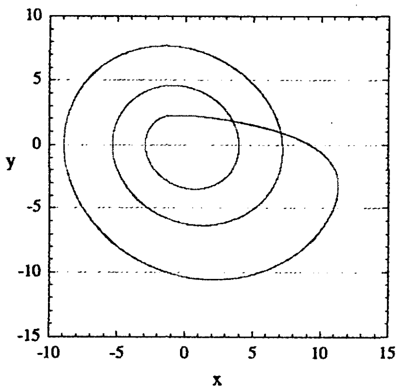

Fig. 20 x-y phase portrait of the Rossler system($x_0 = y_0 = z_0 = 5.5$)

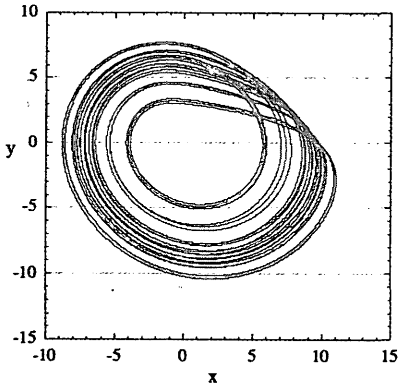

Fig. 21 x-y phase portrait of the Rossler system($x_0 = y_0 = z_0 = 9.5$)

$f = 0.085$ and $f = 0.057$, respectively. For investigating the power spectrum which depends on the frequency, it is confirmed that the period-one,two and three cycle of the Rossler system can be found by employing the NNC with use of hyperbolic tangent function. It is tentatively considired that the hyperbolic tangent function is the best candidate as the threshold function of NNC for controlling Rossler system. The numbers of updating weights in NNC is a good indicator whether the NNC successfully controls the chaotic orbit to the periodic cycle, or not.

Finally, we extend the NNC with three layers such as input layer, hidden layer and output layer. Fig.29 shows that NNC consists of three input neurons, five hidden neurons and three output neurons. The NNC is governed by

$$O_j(t_k) = \sum_{l=1}^{3} W_{jl}(k) X_l(t_k), \tag{40}$$

$$T_j(t_k) = f(O_j(t_k) + h_j(k)), \tag{41}$$

$$Z_i(t_k) = \sum_{j=1}^{5} V_{ij}(k) T_j(t_k), \tag{42}$$

$$U_i(t_k) = f(Z_i(t_k) + \theta_i(k)), \tag{43}$$

where l is the number of input neuron, j, the number of hidden neuron, i, the number of output neuron, W_{jl} is the weight of hidden neuron, V_{ij}, the weight of output neuron, and $O_j(t_k)$ is the signal which goes into neuron of hidden layer through the weight of hidden neuron from lth neuron of input layer, $Z_i(t_k)$ is the signal which goes into neuron of output layer through the wieght of output neuron from jth neuron of hidden layer. $h_j(k)$ is the bias of the jth neuron of the hidden layer, and $\theta_i(k)$ is the bias of the ith neuron of the output layer. The weights and the biases are updated by using

$$\Delta W_{ij}(k) = -\eta X_l T_j(1-T_j)\sum_{i=1}^{3} V_{ij}U_i(1-U_i)\frac{\partial E(k)}{\partial U_i(n_k)}, \tag{44}$$

$$\Delta h_j(k) = -\eta T_j(1-T_j)\sum_{i=1}^{3} V_{ij}U_i(1-U_i)\frac{\partial E(k)}{\partial U_i(n_k)}, \tag{45}$$

$$\Delta V_{ij}(k) = -\eta T_j U_i(1-U_i)\frac{\partial E(k)}{\partial U_i(n_k)}, \tag{46}$$

$$\Delta \theta_i(k) = -\eta U_i(1-U_i)\frac{\partial E(k)}{\partial U_i(n_k)}. \tag{47}$$

For controlling chaos, we applied this method to the Rossler system. But no advantage of this metod is found so far compared with the NNC with two layers. Therefore we tentatively conclude that NNC with two layers and hyperbolic tangent function as the threshold function shows the best peroformance to control the Rossler system. It is also found that the suitable choice of parameter (η, K_C, K_U) which decide network weights are important to stabilize the chaotic orbit.

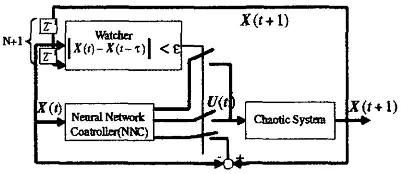
N+1
τ
τ
Watcher
$|X(t)-X(t-\tau)| < \varepsilon$
X(t+1)
X(t)
Neural Network Coatroller(NNC)
U(t)
Chaotic System
X(t+1)
-
+

Fig. 22 Block diagram of the original neural controller

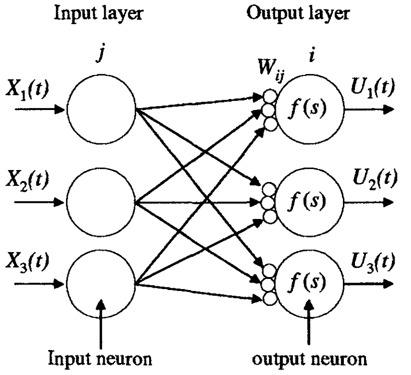

Fig. 23 Neural network controller with two layers

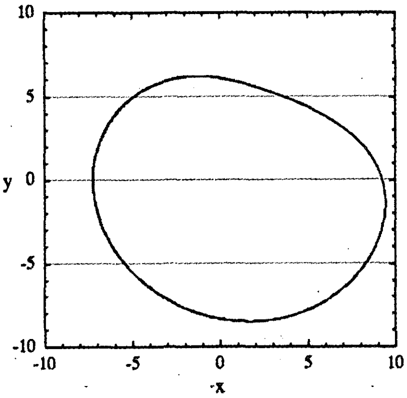

Fig. 24 x-y phase portrait of the Rossler system($\tau = 5.9, t > 100$)

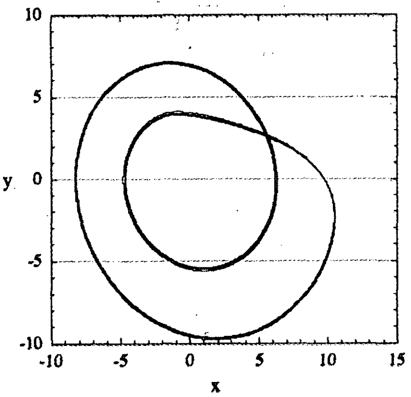

Fig. 25 x-y phase portrait of the Rossler system($\tau = 11.75, t > 100$)

5. Conclusions

The Pyragas method and the on-line trained NNC are tested for the Henon map and the Rossler system. It is found that the desired UPO of Ressler system is stabilized by the delayed feedback controlling method. However, the controllability depends on the choice of the initial condition for period-three cycle as well as the gain of perturbation. The NNC with linear threshold function works well for the control of the chaos in Henon map. It fails in the application to the Rossler system. The NNC is examined by introducing the different threshold function such as sigmoid function and hyperbolic tangent function. We find that the NNC with hyperbolic tangent function shows the best performace among three cases for controlling the Rossler system. It is also found that the suitable choice of parameter (η, k_C, k_U) which decide the network weights are important to stabilize the chaotic orbit. We examine the NNC with three layers. No advantage is found compared with the NNC with two layers for the Rossler system. Once

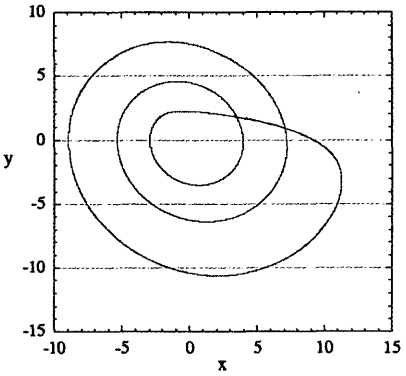

Fig. 26 x-y phase portrait of the Rossler system($\tau = 17.5, t > 100$)

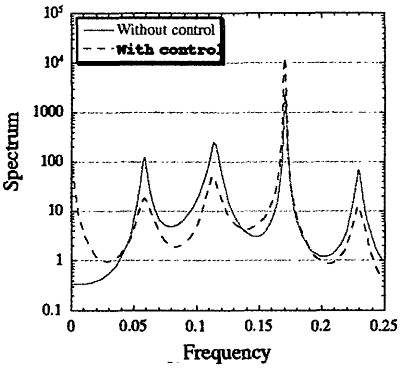

Fig. 27 The dynamics of power spectrum on τ

we find a suitable value of K, then the delayed feedback controlling method is faster than NNC with hyperbolic tangent function, although it is not always possible for the general case.

As a future work, we will test the performance of NNC in which the network weight is evolved by genetic algorithm with good fitness function and compare the

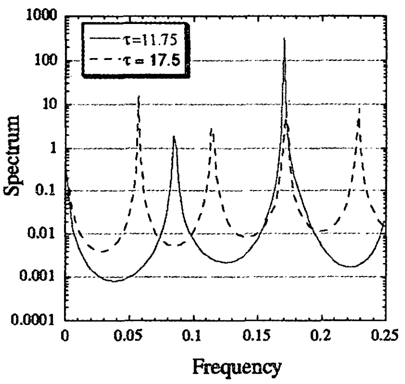

Fig. 28 The dynamics of power spectrum on τ

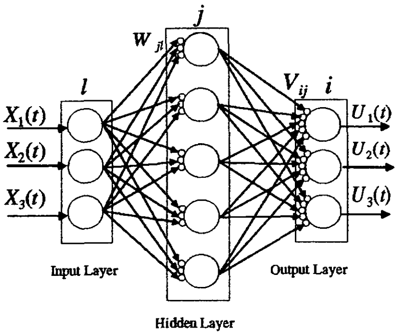

Fig. 29 Neural network controller with three layers

result it with that by the back propagation method.

References

1) P.Y. Chenug and A.Y. Wong: Chaotic behavior and period doubling in plasmas, *Phys. Rev. Lett*, Vol.59, (1987) 551-554

2) T.Fukuyama, K.Taniguchi et al.:Controlling chaos in the current-driven ion acoustic instability, *Phys. Plasma*, Vol.9,(2002) 1570-1574

3) E. Ott, C.Grebogi et al.: Controlling chaos, *Phys. Rev. Lett.*, Vol.64, (1990) 1196-1199

4) K. Pyragas: Continuous control of chaos by self-controlling feedback, *Phys. Lett. A*, Vol.170, (1992) 421-428

5) K. Konishi and H. Kokame: Control of chaotic systems using an on-line trained linear neural controller, *Physica D*, Vol.100, (1997) 423-438

6) R. Weeks and M. Burgess: Evolving artificial neural networks to control chaotic systems, *Phys. Rev. E*, Vol.56, (1997) 1531-1540

7) I.B. Schwartz and I. Triandaf: Tracking unstable orbits in experiments, *Phys, Rev. A*, Vol.46, (1992) 7439-7444

8) P.M. Alsing, A. Gavrielides et al.: Using neural networks for controlling chaos, *Phys. Rev. E*, Vol.49, (1994) 1225-1231

9) K.Konishi and H.Kokame: Stabilizing and tracking unstable focus points in chaotic systems using a neural network, *Phys. Lett. A*, Vol.206, (1997) 203-210

Appendix

A1. Appendix: Symbiotic adaptive neuro-evolution

In this appendix, we apply dsane(Directed Symbiotic Adaptive Neuro-Evolution)[6] to optimize weights of NNC instead of backpropagation method. In this case, the proper choice of the fitness function determines the speed with which the GA can converge on the correct solution.Our goal is to find a neural network that can control a specific system such that the dynamics eventually reaches a fixed point. For simplicity, we consider a period-1 fixed point difined by $X_n = X_{n-1}$, where X is an observable and n is the n-th observation. The fitness function consists of three parts: $F = AF_1 + BF_2 + CF_3$, where A, B and C are adjustable parameters which are system independent. The map to be controlled is iterated 1000 times and the value of each part of the fitness function is set by the behavior of the map and the network.

$F_1 \equiv 1 - \langle \Delta_n \rangle$ indicates whether the network has stabilized a fixed point by interations. Δ_n is defined by $\equiv |X_n - X_{n-1}|/S$ and S is the size of the uncontrolled attractor of the system. The average $\langle \cdots \rangle$ is taken over the last 40 iterations of the map. The small values of Δ_n result in a larger fitness F_1.

$F_2 \equiv 1 - ln(\lambda)$ quantifies the growth rate of Δ_n near the fixed point. If Δ_n is smaller than 0.01, the values of Δ_n are stored until Δ_n grows larger than 0.10. λ is the geometric mean of the quantities Δ_{n+1}/Δ_n for all Δ_n which have been stored. When the fixed point is successfuly stabilized by the neural network, $\lambda \sim 1$.

F_3 rewards networks which are optimal controllers. Randomly chosen network weights lead to a network which applies the largest possible perturbation δP to the system. F_3 is defined by the fraction of iterations for which $|\delta P|$ is smaller than $0.95\delta P_{max}$, where δP_{max} is the magnitude of the maximum allowed perturbation. If the system is exactly on the fixed point, the necessary perturbation is $\delta P = 0$.

NNC consists of four input neurons, seven hidden neurons, and one output neurons. The input neurons are assigned the values X_n, X_{n-1}, X_{n-2} and X_{n-3}. The value y of the output neuron sets the perturbation applied to the system $\delta P = \delta P_{max}(2y - 1)$, where $0 < y < 1$ and $|\delta P| < \delta P_{max}$. Parameters for the evolution in the GA are given as follows. (1) population size consists of 100 neurons, (2) neurons in 30 percent of the population is preserved for each generation, (3) neurons in 60 percent of the populations is formed by crossover, and (4) neurons in 10 percent of the population is formed by mutation.

We have translated the original code written in C into Fortran 90. The benchmark of two codes are done, however, it is found that GA depends on the random number(in other words, it is the probabilistic), therefore, both results are not agreed with each other. Some numerical results for Henon map are shown. The map is described by

$$\begin{cases} X_1(n+1) = 1.29 + 0.3X_2(n) - X_1(n)^2 + \epsilon + \delta P_1, \\ X_2(n+1) = X_1(n) + \delta P_2. \end{cases} \tag{A1}$$

where $\epsilon = 0.001$ is the noiselevel.

We have three runs. Fig.A1 shows the evolution of the fitness in each generation for the Henon map. It is shown that the evolution of fitness depends on initial populations, mutation and crossover which are controlled by random number. Two cases successfully evoluve to high fitness values, however, one case fails. Fig.A2 shows performance of the best network in cases with fitness values: 600,800,850 and 900, respectively. The Henon map is well controlled for the case with the fitness value larger than 900.

As a future work, this algorithm should be applied to the Rossler system to evaluate the network performace.

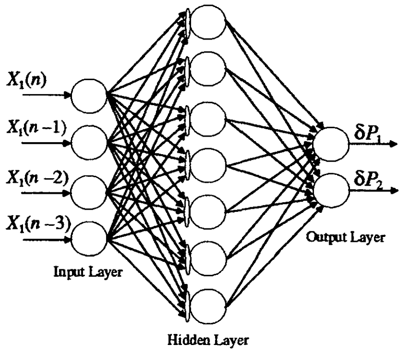

Fig. A1 Neural network controller

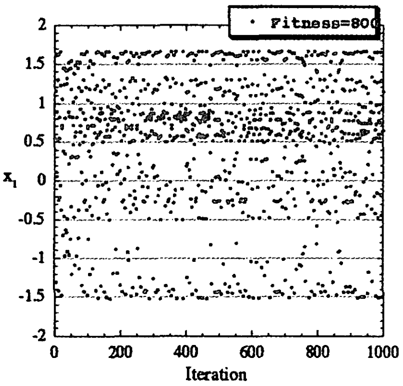

Fig. A4 Iteration of x for Henon map

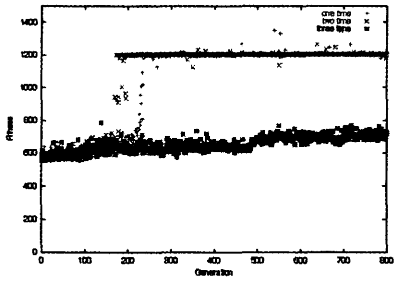

Fig. A2 History of the evolution for the Henon map

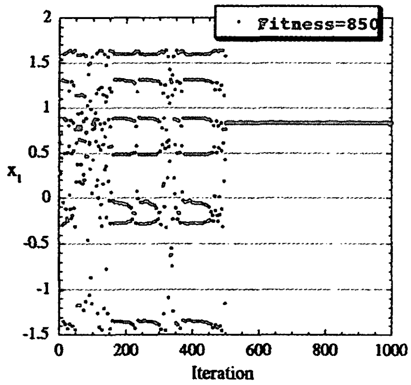

Fig. A5 Iteration of x for Henon map

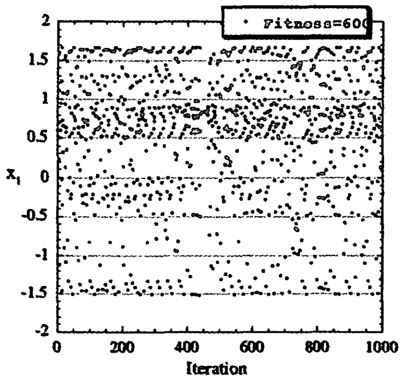

Fig. A3 Iteration of x for Henon map

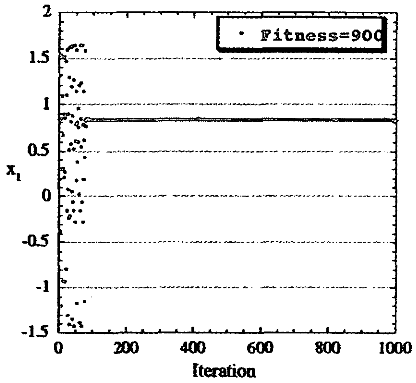

Fig. A6 Iteration of x for Henon map