We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



14

An Improved Extremum Seeking Algorithm Based on the Chaotic Annealing Recurrent Neural Network and Its Application

Yun-an Hu, Bin Zuo and Jing Li
Department of Control Engineering, Naval Aeronautical and Astronautical University
P. R. China

1. Introduction

Extremum seeking problem deals with the problem of minimizing or maximizing a plant over a set of decision variables[1]. Extremum seeking problems represent a class of widespread optimization problems arising in diverse design and planning contexts. Many large-scale and real-time applications, such as traffic routing and bioreactor systems, require solving large-scale extremum seeking problem in real time. In order to solve this class of extremum seeking problems, a novel extremum seeking algorithm was proposed in the 1950's. Early work on performance improvement by extremum seeking can be found in Tsien. In the 1950s and the 1960s, Extremum seeking algorithm was considered as an adaptive control method[2]. Until 1990s sliding mode control for extremum seeking has not been utilized successfully[3]. Subsequently, a method of adding compensator dynamics in ESA was proposed by Krstic, which improved the stability of the controlled extremum control system[4]. Although those methods improved tremendously the performance of ESA, the "chatter" problem of the output and the switching of the control law and incapability of escaping from the local minima limit the application of ESA.

In order to solve those problems in the conventional ESA and improve the capability of global searching, an improved chaotic annealing recurrent neural network (CARNN) is proposed in the paper. The method of introducing a chaotic annealing recurrent neural network into ESA is proposed in the paper. First, an extremum seeking problem is converted into the process of seeking the global extreme point of the plant where the slope of cost function is zero. Second, an improved CARNN is constructed; and then we can apply the CARNN to finding the global extreme point and stabilizing the plant at that point. The CARNN proposed in the paper doesn't make use of search signals such as sinusoidal periodic signals, so the method can solve the "chatter" problem of the output and the switching of the control law in the general ESA and improve the dynamic performance of extremum seeking system. At the same time, CARNN utilizes the randomicity and the property of global searching of chaos system to improve the capability of global searching of the system[5-6], During the process of optimization, chaotic annealing is realized by decaying the amplitude of the chaos noise and the accepting probability continuously. Adjusting the probability of acceptance could influence the rate of convergence. The process of optimization was divided into two phases: the coarse search based on chaos and the

Open Access Database www.i-techonline.com

Source: Recurrent Neural Networks, Book edited by: Xiaolin Hu and P. Balasubramaniam, ISBN 978-953-7619-08-4, pp. 400, September 2008, I-Tech, Vienna, Austria

elaborate search based on RNN. At last, CARNN will stabilize the system to the global extreme point, which is validated by simulating a simplified UAV tight formation flight model and a typical Schaffer Function. At the same time, the stability analysis of ESA can be simplified by the proposed method.

2. Annealing recurrent neural network description

2.1 Problem formulation

Consider a general nonlinear system:

$$\begin{aligned} \dot{x} &= f\left(x(t),u(t)\right) \\ y &= F\left(x(t)\right) \end{aligned} \tag{1}$$

Where $x \in R^n, u \in R^m$ and $y \in R$ are the states, the system inputs and the system output, respectively. $F(x)$ is also defined as the cost function of the system (1). $f(x,u)$ and $F(x)$ are smooth functions. If the nonlinear system (1) is an extremum seeking system, then it must satisfy the following assumptions.

Assumption 1: There is a smooth control law[7]:

$$u(t) = \alpha\left(x(t),\theta\right) \tag{2}$$

to stabilize the nonlinear system(1), where $\theta=\left[\theta_1,\theta_2,\cdots,\theta_i,\cdots,\theta_p\right]^T \left(i \in [1,2,\cdots,p]\right)$ is a parameter vector of p dimension which determines a unique equilibrium vector.

With the control law (2), the closed-loop system of the nonlinear system (1) can be written as:

$$\dot{x} = f\left(x,\alpha\left(x,\theta\right)\right)$$

Assumption 2: There is a smooth function $x_e : R^p \to R^n$ such that:

$$f\left(x,\alpha\left(x,\theta\right)\right) = 0 \leftrightarrow x = x_e\left(\theta\right)$$

Assumption 3: The static performance map at the equilibrium point $x_e(\theta)$ from θ to y represented by:

$$y = F\left(x_e\left(\theta\right)\right) = F\left(\theta\right) \tag{3}$$

is smooth and has a unique global maximum or minimum vector $\theta^* \in R^p, \theta^* = \left[\theta_1^*,\theta_2^*,\cdots,\theta_p^*\right]^T$ such that:

$$\frac{\partial F\left(\theta^*\right)}{\partial \theta_i} = 0, \left(i = 1,2,\cdots,p\right)$$

at the same time $\dfrac{\partial^2 F\left(\theta^*\right)}{\partial \theta_i^2} < 0$ or $\dfrac{\partial^2 F\left(\theta^*\right)}{\partial \theta_i^2} > 0$

Differentiating (3) with respect to time yields the relation between $\dot{\theta}$ and $\dot{y}(t)$.

$$\partial(\theta(t))\dot{\theta}(t) = \dot{y}(t) \tag{4}$$

Where $\partial(\theta(t)) = \left[\frac{\partial F(\theta)}{\partial \theta_1}, \frac{\partial F(\theta)}{\partial \theta_2}, \cdots, \frac{\partial F(\theta)}{\partial \theta_p}\right]^T$ and $\dot{\theta}(t) = \left[\dot{\theta}_1, \dot{\theta}_2, \cdots, \dot{\theta}_p\right]^T$.

Based on Assumption 3, once the seeking vector θ of the extremum seeking system (1) converges to the global extreme vector θ^*, then $|\partial(\theta)| = \left[\left|\frac{\partial F(\theta)}{\partial \theta_1}\right|, \left|\frac{\partial F(\theta)}{\partial \theta_2}\right|, \cdots, \left|\frac{\partial F(\theta)}{\partial \theta_p}\right|\right]^T$ must also converge to zero. A CARNN is introduced into ESA in order to minimize $|\partial(\theta)|$ in finite time. Certainly the system (1) is subjected to (4).
Then, the extremum seeking problem can be written as follows.

$$\theta^* = arg\,min_{\theta \in \hat{\theta}} |\partial(\theta)|$$

$$\text{Subject to: } \partial^T(\theta(t))\dot{\theta}(t) = \dot{y}(t) \tag{5}$$

The optimization (5) is then equivalent to

$$\text{Minimize: } f_1(\upsilon) = c^T \upsilon$$

$$\text{Subject to: } p_1(\upsilon) = A\upsilon - b = 0 \tag{6}$$

Where, $\partial^T(\theta)$ denotes the transpose of $\partial(\theta).\upsilon = \left[\partial(\theta) \quad |\partial(\theta)| \quad \dot{\theta}(t)\right]^T$, $c = \left[0_{1\times p} \quad 1_{1\times p} \quad 0_{1\times p}\right]^T$, $A = \begin{bmatrix} 1_{1\times p} & -sign(\partial^T(\theta)) & 0_{1\times p} \\ \dot{\theta}^T(t) & 0_{1\times p} & 0_{1\times p} \\ 0_{1\times p} & 0_{1\times p} & \partial^T(\theta) \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ \dot{y}(t) \\ \dot{y}(t) \end{bmatrix}$, and $\text{sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$.

By the dual theory, the dual program corresponding to the program (6) is

$$\text{Maximize: } f_2(\omega) = b^T \omega$$

$$\text{Subject to: } p_2(\omega) = A^T \omega - c = 0 \tag{7}$$

Where, ω denotes the dual vector of υ, $\omega^T = \left[\omega_1 \quad \omega_2 \quad \omega_3\right]_{1\times 3}$.
Therefore, an extremum seeking problem is converted into the programs defined in (6) and (7).

2.2 Annealing Recurrent Neural Network (ARNN) design

In view of the primal and dual programs (6) and (7), define the following energy function:

$$E(\upsilon, \omega) = \frac{1}{2}T(t)\left(f_1(\upsilon) - f_2(\omega)\right)^2 + \frac{1}{2}\left\|p_1(\upsilon)\right\|^2 + \frac{1}{2}\left\|p_2(\omega)\right\|^2 \tag{8}$$

Clearly, the energy function (8) is convex and continuously differentiable. The first term in (8) is the squared of the difference between two objective functions of the programs (6) and (7), respectively. The second and the third terms are for the equality constraints of (6) and (7). $T(t)$ denotes a time-varying annealing parameter.

With the energy function defined in (8), the dynamics for CARNN solving (6) and (7) can be defined by the negative gradient of the energy function as follows.

$$\frac{d\sigma}{dt} = -\mu \nabla E(\sigma) \tag{9}$$

Where, $\sigma = (\upsilon, \omega)^T$, $\nabla E(\sigma)$ is the gradient of the energy function $E(\sigma)$ defined in (9), and μ is a positive scalar constant, which is used to scale the convergence rate of annealing recurrent neural network.

The dynamical equation (9) of annealing recurrent neural network can be expressed as:

$$\frac{du_1}{dt} = -\mu \frac{\partial E(\upsilon,\omega)}{\partial \upsilon} = -\mu \left[T(t) \cdot \frac{\partial f_1(\upsilon)}{\partial \upsilon} \cdot \left(f_1(\upsilon) - f_2(\omega) \right) + \frac{\partial p_1(\upsilon)}{\partial \upsilon} \cdot p_1(\upsilon) \right]$$

$$= -\mu \left[T(t) c \left(c^T \upsilon - b^T \omega \right) + A^T \left(A\upsilon - b \right) \right] \tag{10}$$

$$\frac{du_2}{dt} = -\mu \frac{\partial E(\upsilon,\omega)}{\partial \omega} = -\mu \left[-T(t) \cdot \frac{\partial f_2(\omega)}{\partial \omega} \cdot \left(f_1(\upsilon) - f_2(\omega) \right) + \frac{\partial p_2(\omega)}{\partial \omega} \cdot p_2(\omega) \right]$$

$$= -\mu \left[-T(t) b \left(c^T \upsilon - b^T \omega \right) + A \left(A^T \omega - c \right) \right] \tag{11}$$

$$\upsilon = q(u_1) \tag{12}$$

$$\omega = q(u_2) \tag{13}$$

Where, $q(\ )$ is a sigmoid activation function, $\upsilon = q(u_1) = \frac{b_1 - a_1}{1 + e^{-u_1/\varepsilon_1}} + a_1$ and $\omega = q(u_2) = \frac{b_2 - a_2}{1 + e^{-u_2/\varepsilon_2}} + a_2$. a_1 and b_1 denote the upper bound and the below bound of υ. a_2 and b_2 denote the upper bound and the below bound of ω. $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$.

The annealing recurrent neural network is described as the equations (10) ~ (13), which are determined by the number of decision variables such as (υ, ω), (u_1, u_2) is the column vector of instantaneous net inputs to neurons, (υ, ω) is the column output vector of neurons. The lateral connection weight matrix is defined as $\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} -\mu\left(T(t) cc^T + A^T A\right) & \mu T(t) cb^T \\ \mu T(t) bc^T & -\mu\left(T(t) bb^T + AA^T\right) \end{bmatrix}$, the biasing threshold vector of the neurons is defined as $\begin{bmatrix} \vartheta_1 \\ \vartheta_2 \end{bmatrix} = \begin{bmatrix} \mu A^T b \\ \mu Ac \end{bmatrix}$.

3. Convergence analysis

In this section, analytical results on the stability of the proposed annealing recurrent neural network and feasibility and optimality of the steady-state solutions to the programs described in (6) and (7) are presented.

3.1 Solution feasibility

Theorem 1. Assume that the Jacobian matrices $J\left[q\left(u_1\right)\right]$ and $J\left[q\left(u_2\right)\right]$ exist and are positive semidefinite. If the temperature parameter $T(t)$ is nonnegative, strictly monotone decreasing for $t \geq 0$, and approaches zero as time approaches infinity, then the annealing recurrent neural network (10) ~ (13) is asymptotically stable.

Proof: Consider the following Lyapunov function.

$$L = E(\upsilon,\omega) = \frac{1}{2}T(t)\left(f_1(\upsilon) - f_2(\omega)\right)^2 + \frac{1}{2}\left\|p_1(\upsilon)\right\|^2 + \frac{1}{2}\left\|p_2(\omega)\right\|^2 \tag{14}$$

Apparently, $L(t) > 0$. The difference of L along time trajectory of (14) is as follows:

$$\frac{dL}{dt} = T(t)\left[\frac{\partial f_1(\upsilon)}{\partial \upsilon}\cdot\frac{d\upsilon}{dt} - \frac{\partial f_2(\omega)}{\partial \omega}\cdot\frac{d\omega}{dt}\right]\left(f_1(\upsilon) - f_2(\omega)\right) + \frac{\partial p_1(\upsilon)}{\partial \upsilon}\cdot\frac{d\upsilon}{dt}\cdot p_1(\upsilon)$$

$$+\frac{\partial p_2(\omega)}{\partial \omega}\cdot\frac{d\omega}{dt}\cdot p_2(\omega) + \frac{1}{2}\frac{dT(t)}{dt}\left(f_1(\upsilon) - f_2(\omega)\right)^2$$

$$= \left[T(t)\cdot\frac{\partial f_1(\upsilon)}{\partial \upsilon}\cdot\left(f_1(\upsilon) - f_2(\omega)\right) + \frac{\partial p_1(\upsilon)}{\partial \upsilon}\cdot p_1(\upsilon)\right]\cdot\frac{d\upsilon}{dt}$$

$$+\left[-T(t)\cdot\frac{\partial f_2(\omega)}{\partial \omega}\cdot\left(f_1(\upsilon) - f_2(\omega)\right) + \frac{\partial p_2(\omega)}{\partial \omega}\cdot p_2(\omega)\right]\cdot\frac{d\omega}{dt}$$

$$+\frac{1}{2}\frac{dT(t)}{dt}\left(f_1(\upsilon) - f_2(\omega)\right)^2 \tag{15}$$

According to the equations (10) and (11), and the following equations

$$\frac{d\upsilon}{dt} = J\left[q\left(u_1\right)\right]\cdot\frac{du_1}{dt} \tag{16}$$

$$\frac{d\omega}{dt} = J\left[q\left(u_2\right)\right]\cdot\frac{du_2}{dt} \tag{17}$$

We can have:

$$\frac{dL}{dt} = -\frac{1}{\mu}\cdot\frac{du_1}{dt}\cdot J\left[q(u_1)\right]\cdot\frac{du_1}{dt} - \frac{1}{\mu}\cdot\frac{du_2}{dt}\cdot J\left[q(u_2)\right]\cdot\frac{du_2}{dt} + \frac{1}{2}\frac{dT(t)}{dt}\left(f_1(\upsilon) - f_2(\omega)\right)^2 \tag{18}$$

We know that the Jacobian matrices of $J\left[q(u_1)\right]$ and $J\left[q(u_2)\right]$ both exist and are positive semidefinite and μ is a positive scalar constant. If the time-varying annealing parameter $T(t)$ is nonnegative, strictly monotone decreasing for $t \geq 0$, and approaches zero as time approaches infinity, then dL/dt is negative definite. Because $T(t)$ represents the annealing effect, the simple examples of $T(t)$ can described as follows.

$$T(t) = \beta\alpha^{-\eta t} \tag{19}$$

$$T(t) = \beta(1+t)^{-\eta} \tag{20}$$

Where $\alpha > 1$, $\beta > 0$ and $\eta > 0$ are constant parameters. Parameters β and η can be used to scale the annealing parameter.

Because $L(t)$ is positive definite and radially unbounded, and dL/dt is negative definite. According to the Lyapunov's theorem, the designed annealing recurrent neural network is asymptotically stable.

Theorem 2. Assume that the Jacobian matrices $J\left[q(u_1)\right]$ and $J\left[q(u_2)\right]$ exist and are positive semidefinite. If $T(t) \geq 0$, $dT(t)/dt < 0$ and $\lim_{t\to\infty} T(t) = 0$, then the steady state of the annealing neural network represents a feasible solution to the programs described in equations (6) and (7).

Proof: The proof of Theorem 1 shows that the energy function $E(\upsilon,\omega)$ is positive definite and strictly monotone decreasing with respect to time t, which implies $\lim_{t\to\infty} E\left(\upsilon,\omega,T(t)\right) = 0$. Because $\lim_{t\to\infty} T(t) = 0$, then we have

$$\lim_{t\to\infty} E\left(\upsilon,\omega,T(t)\right) = \lim_{t\to\infty}\left(\frac{1}{2}T(t)\left(f_1(\upsilon) - f_2(\omega)\right)^2 + \frac{1}{2}\left\|p_1(\upsilon)\right\|^2 + \frac{1}{2}\left\|p_2(\omega)\right\|^2\right) \tag{21}$$

$$= \lim_{t\to\infty}\left(\frac{1}{2}\left\|p_1\left(\upsilon(t)\right)\right\|^2 + \frac{1}{2}\left\|p_2\left(\omega(t)\right)\right\|^2\right) = 0 \tag{22}$$

Because $p_1\left(\upsilon(t)\right)$ and $p_2\left(\omega(t)\right)$ are continuous, $\lim_{t\to\infty}\left(\frac{1}{2}\left\|p_1\left(\upsilon(t)\right)\right\|^2 + \frac{1}{2}\left\|p_2\left(\omega(t)\right)\right\|^2\right) = \frac{1}{2}\left\|p_1\left(\lim_{t\to\infty}\upsilon(t)\right)\right\|^2$ $+\frac{1}{2}\left\|p_2\left(\lim_{t\to\infty}\omega(t)\right)\right\|^2 = \frac{1}{2}\left\|p_1(\bar{\upsilon})\right\|^2 + \frac{1}{2}\left\|p_2(\bar{\omega})\right\|^2 = 0$, so we have $p_1(\bar{\upsilon}) = 0$ and $p_2(\bar{\omega}) = 0$, where $\bar{\upsilon}$ and $\bar{\omega}$ are the stable solutions of υ and ω.

3.2 Solution optimality

Firstly, Let $F_1(\upsilon) = \begin{bmatrix}(f_1(\upsilon))\\(f_1(\upsilon))\\(f_1(\upsilon))\end{bmatrix} = I_{3\times1}\cdot\left(f_1(\upsilon)\right)$ and $F_2(\omega) = \begin{bmatrix}(f_2(\omega))\\(f_2(\omega))\\(f_2(\omega))\end{bmatrix} = I_{3\times1}\cdot\left(f_2(\omega)\right)$ be the augmented vector.

Theorem 3. Assume that the Jacobian matrices $J\left[q\left(u_1\right)\right] \neq 0$ and $J\left[q\left(u_2\right)\right] \neq 0$ and are positive semidefinite, $\forall t \geq 0$, and $\nabla\left(f_1(\upsilon)\right) \neq 0$ and $\nabla\left(f_2(\omega)\right) \neq 0$. If $dT(t)/dt < 0$, $\lim_{t\to\infty} T(t) = 0$ and

$$T(t) \geq \max\left\{0, \frac{\left(\nabla p_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial p_1(\upsilon)}{\partial \upsilon} p_1(\upsilon) - \nabla F_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial p_1(\upsilon)}{\partial \upsilon} p_1(\upsilon)\right)}{\left(\nabla F_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial f_1(\upsilon)}{\partial \upsilon}\left(f_1(\upsilon) - f_2(\omega)\right) - \nabla p_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial f_1(\upsilon)}{\partial \upsilon}\left(f_1(\upsilon) - f_2(\omega)\right)\right)}, \right.$$

$$\left. \frac{\left(\nabla p_2\left[\omega(t)\right]^T J\left[q(u_2)\right]\frac{\partial p_2(\omega)}{\partial \omega} p_2(\omega) - \nabla F_2\left[\omega(t)\right]^T J\left[q(u_2)\right]\frac{\partial p_2(\omega)}{\partial \omega} p_2(\omega)\right)}{\left(\nabla F_2\left[\omega(t)\right]^T J\left[q(u_2)\right]\frac{\partial f_2(\omega)}{\partial \omega}\left(f_1(\upsilon) - f_2(\omega)\right) - \nabla p_2\left[\omega(t)\right]^T J\left[q(u_2)\right]\frac{\partial f_2(\omega)}{\partial \omega}\left(f_1(\upsilon) - f_2(\omega)\right)\right)} \right\} \quad (23)$$

then the steady states $\bar{\upsilon}$ and $\bar{\omega}$ of the annealing neural network represents the optimal solutions υ^* and ω^* to the programs described in equations (6) and (7).

Proof: According to the equation (23), we know

$$T(t) \geq \frac{\left(\nabla p_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial p_1(\upsilon)}{\partial \upsilon} p_1(\upsilon) - \nabla F_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial p_1(\upsilon)}{\partial \upsilon} p_1(\upsilon)\right)}{\left(\nabla F_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial f_1(\upsilon)}{\partial \upsilon}\left(f_1(\upsilon) - f_2(\omega)\right) - \nabla p_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial f_1(\upsilon)}{\partial \upsilon}\left(f_1(\upsilon) - f_2(\omega)\right)\right)}$$

The above equation implies

$$\mu T(t)\left[\nabla F_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right] - \nabla p_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\right]\frac{\partial f_1(\upsilon)}{\partial \upsilon}\left(f_1(\upsilon) - f_2(\omega)\right)$$

$$\geq \mu\nabla p_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial p_1(\upsilon)}{\partial \upsilon} p_1(\upsilon) - \mu\nabla F_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial p_1(\upsilon)}{\partial \upsilon} p_1(\upsilon)$$

Rearranging the above inequality, we have

$$\mu T(t)\nabla F_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial f_1(\upsilon)}{\partial \upsilon}\left(f_1(\upsilon) - f_2(\omega)\right) + \mu\nabla F_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial p_1(\upsilon)}{\partial \upsilon} p_1(\upsilon)$$

$$-\mu T(t)\nabla p_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial f_1(\upsilon)}{\partial \upsilon}\left(f_1(\upsilon) - f_2(\omega)\right) - \mu\nabla p_1\left[\upsilon(t)\right]^T J\left[q(u_1)\right]\frac{\partial p_1(\upsilon)}{\partial \upsilon} p_1(\upsilon) \geq 0$$

That is also

$$\mu\nabla F_1\left[\upsilon(t)\right]^T \cdot J\left[q(u_1)\right] \cdot \left[\left(T(t) \cdot \frac{\partial f_1(\upsilon)}{\partial \upsilon} \cdot \left(f_1(\upsilon) - f_2(\omega)\right) + \frac{\partial p_1(\upsilon)}{\partial \upsilon} \cdot p_1(\upsilon)\right)\right]$$

$$-\mu\nabla p_1\left[\upsilon(t)\right]^T \cdot J\left[q(u_1)\right]\cdot\left[\left(T(t)\cdot\frac{\partial f_1(\upsilon)}{\partial\upsilon}\cdot\left(f_1(\upsilon)-f_2(\omega)\right)+\frac{\partial p_1(\upsilon)}{\partial\upsilon}\cdot p_1(\upsilon)\right)\right]\geq 0$$

Substituting equations (10), (11), (16) and (17) into the above inequality, we have

$$\nabla F_1\left[\upsilon(t)\right]^T \cdot J\left[q(u_1)\right]\cdot\frac{du_1(t)}{dt}-\nabla p_1\left[\upsilon(t)\right]^T \cdot J\left[q(u_1)\right]\cdot\frac{du_1(t)}{dt}=$$

$$\nabla F_1\left[\upsilon(t)\right]^T \cdot\frac{d\upsilon(t)}{dt}-\nabla p_1\left[\upsilon(t)\right]^T \cdot\frac{d\upsilon(t)}{dt}=\frac{dF_1(\upsilon(t))}{dt}-\frac{dp_1(\upsilon(t))}{dt}\leq 0$$

Therefore, we have $dF_1(\upsilon(t))/dt \leq dp_1(\upsilon(t))/dt$, which implies $F_1(\upsilon(t''))-F_1(\upsilon(t'))\leq p_1(\upsilon(t''))-p_1(\upsilon(t'))$ for any $t'\leq t''$. Let t^* be the time associated with an optimal solution υ^*. We have $F_1(\upsilon(\infty))-F_1(\upsilon(t^*))\leq p_1(\upsilon(\infty))-p_1(\upsilon(t^*))$; that is $F_1(\overline{\upsilon})-F_1(\upsilon^*)\leq p_1(\overline{\upsilon})-p_1(\upsilon^*)$. Because $p_1(\overline{\upsilon})=p_1(\upsilon^*)=0$, $F_1(\overline{\upsilon})\leq F_1(\upsilon^*)$. At last, we have $f_1(\overline{\upsilon})\leq f_1(\upsilon^*)$. Also, because $\upsilon^*=\arg\min_{\upsilon\in\hat{V}} f_1(\upsilon)$, $f_1(\overline{\upsilon})\geq f_1(\upsilon^*)$ by definition of υ^*. Consequently, $f_1(\overline{\upsilon})=f_1(\upsilon^*)=\min_{\upsilon\in\hat{V}} f_1(\upsilon)$, where $\hat{V}$ denotes the feasible region of the optimal solution υ^*.

Next, according to the equation (23), we also know

$$T(t)\geq\frac{\left(\nabla p_2\left[\omega(t)\right]^T J\left[q(u_2)\right]\frac{\partial p_2(\omega)}{\partial\omega}p_2(\omega)-\nabla F_2\left[\omega(t)\right]^T J\left[q(u_2)\right]\frac{\partial p_2(\omega)}{\partial\omega}p_2(\omega)\right)}{\left(\nabla F_2\left[\omega(t)\right]^T J\left[q(u_2)\right]\frac{\partial f_2(\omega)}{\partial\omega}\left(f_1(\upsilon)-f_2(\omega)\right)-\nabla p_2\left[\omega(t)\right]^T J\left[q(u_2)\right]\frac{\partial f_2(\omega)}{\partial\omega}\left(f_1(\upsilon)-f_2(\omega)\right)\right)}$$

By the same reasoning, we know $\frac{dF_2(\omega(t))}{dt}\geq\frac{dp_2(\omega(t))}{dt}$, which implies $F_2(\omega(t''))-F_2(\omega(t'))$ $\geq p_2(\omega(t''))-p_2(\omega(t'))$ for any $t'\leq t''$. Let t^* be the time associated with an optimal solution ω^*. We have $F_2(\omega(\infty))-F_2(\omega(t^*))\geq p_2(\omega(\infty))-p_2(\omega(t^*))$; that is $F_2(\overline{\omega})-F_2(\omega^*)\geq p_2(\overline{\omega})-p_2(\omega^*)$. Because $p_2(\overline{\omega})=p_2(\omega^*)=0$, $F_2(\overline{\omega})\geq F_2(\omega^*)$. At last, we have $f_2(\overline{\omega})\geq f_2(\omega^*)$. Also, because $\omega^*=\arg\max_{\omega\in\hat{U}} f_2(\omega)$, $f_2(\omega^*)\geq f_2(\overline{\omega})$ by definition of ω^*. Consequently, $f_2(\overline{\omega})=f_2(\omega^*)=\max_{\omega\in\hat{U}} f_2(\omega)$, where $\hat{U}$ denotes the feasible region of the optimal solution ω^*.

4. A chaotic annealing recurrent neural network description

In order to improve the global searching performance of the designed annealing recurrent neural network, we introduce chaotic factors into the designed neural network. Therefore, the structure of a chaotic annealing recurrent neural network is described as follows.

$$\frac{du_1}{dt}=\begin{cases}-\mu\left[T(t)\cdot\dfrac{\partial f_1(\upsilon)}{\partial\upsilon}\cdot\left(f_1(\upsilon)-f_2(\omega)\right)+\dfrac{\partial p_1(\upsilon)}{\partial\upsilon}\cdot p_1(\upsilon)\right]+\eta_1\left(\chi_1(b_1-a_1)+a_1\right) & random<P_1(t)\\ -\mu\left[T(t)\cdot\dfrac{\partial f_1(\upsilon)}{\partial\upsilon}\cdot\left(f_1(\upsilon)-f_2(\omega)\right)+\dfrac{\partial p_1(\upsilon)}{\partial\upsilon}\cdot p_1(\upsilon)\right] & \text{otherwise}\end{cases} \tag{24}$$

$$\frac{du_2}{dt}=\begin{cases}-\mu\left[-T(t)\cdot\dfrac{\partial f_2(\omega)}{\partial\omega}\cdot\left(f_1(\upsilon)-f_2(\omega)\right)+\dfrac{\partial p_2(\omega)}{\partial\omega}\cdot p_2(\omega)\right]+\eta_2\left(\chi_2(b_2-a_2)+a_2\right) & random<P_2(t)\\ -\mu\left[-T(t)\cdot\dfrac{\partial f_2(\omega)}{\partial\omega}\cdot\left(f_1(\upsilon)-f_2(\omega)\right)+\dfrac{\partial p_2(\omega)}{\partial\omega}\cdot p_2(\omega)\right] & \text{otherwise}\end{cases} \tag{25}$$

$$\upsilon=q(u_1)=\frac{b_1-a_1}{1+e^{-u_1/\varepsilon_1}}+a_1 \tag{26}$$

$$\omega=q(u_2)=\frac{b_2-a_2}{1+e^{-u_2/\varepsilon_2}}+a_2 \tag{27}$$

$$\eta_i(t+1)=(1-\kappa)\eta_i(t)\ \ i=1,2 \tag{28}$$

$$P_i(t+1)=\begin{cases}P_i(t)-\delta & P_i(t)>0\\ 0 & \text{otherwise}\end{cases} \tag{29}$$

$$\chi_i(t+1)=\gamma\chi_i(t)\left(1-\chi_i(t)\right) \tag{30}$$

Where $\gamma=4$, $P_i(0)>0$, $0<\kappa<1$, $0<\delta<1$, $\eta_i(0)>0$, $\varepsilon_1>0$ and $\varepsilon_2>0$. We know that equation (30) is a Logistic map, when $\gamma=4$, the chaos phenomenon will happen in the system.

As time approaches infinity, the chaotic annealing recurrent neural network will evolve into the annealing recurrent neural network (10) ~ (13). Therefore, we must not repeatedly analyze the stability and solution feasibility and solution optimality of the chaotic annealing recurrent neural network (24) ~ (30).

5. Simulation analysis

5.1 A simplified tight formation flight model simulation

Consider a simplified tight formation flight model consisting of two Unmanned Aerial Vehicles[8].

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0\\ -20 & -9 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & -35 & -15\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}+\begin{bmatrix}0 & 0\\ 1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix} \tag{31}$$

with a cost function given by

$$y(t) = -10\left(x_1(t) + 0\right)^2 - 5\left(x_3(t) + 9\right)^2 + 590 \tag{32}$$

Where x_1 is the vertical separation of two Unmanned Aerial Vehicles, x_2 is the differential of x_1, x_3 is the lateral separation of two Unmanned Aerial Vehicles, x_4 is the differential of x_3 and y is the upwash force acting on the wingman. It is clear that the global maximum point is $x_1^* = 0$ and $x_3^* = -9$, where the cost function $y(t)$ reaches its maximum $y^* = 590$.

A control law based on sliding mode theory is given by:

$$\begin{cases} \sigma_1 = s_1 x_1 + x_2 \\ u_1 = 20x_1 + (9 - s_1)x_2 - k_1 sign(\sigma_1 - s_1\theta_1) \\ \sigma_2 = s_2 x_3 + x_4 \\ u_2 = 35x_3 + (15 - s_2)x_4 - k_2 sign(\sigma_2 - s_2\theta_2) \end{cases} \tag{33}$$

Where σ_1, σ_2 are two sliding mode surfaces, s_1, k_1, s_2, k_2 are positive scalar constants, $\theta = [\theta_1, \theta_2]$ are an extremum seeking vector.

Remark: The control law is given in (33), which is based on sliding mode theory. We choose $sign(\sigma_i - s_i\theta_i)$, $(i = 1,2)$ so that x_1 and x_3 entirely traces θ_1 and θ_2 in the sliding mode surfaces respectively, and the system will be stable at θ_1^* and θ_2^* finally.

The initial conditions of the system (31) are given as $x_1(0) = -2$, $x_2(0) = 0$, $x_3(0) = -4$, $x_4(0) = 0$, $\theta_1(0) = -2$, $\theta_2(0) = -4$. Choose $T(t) = \beta\alpha^{-\eta t}$, where $\beta = 0.01$, $\alpha = e$, $\eta = 5$. Applying CARNN to system (31), the parameters are given as: $\mu = 23.5$, $\gamma = 4$, $P_1(0) = P_2(0) = 1$, $\kappa = 0.01$, $\delta = 0.01$, $\varepsilon_1 = 10$, $\varepsilon_2 = 10$, $\chi_1(0) = 0.912$, $\chi_2(0) = 0.551$, $\eta_1(0) = [-10 \quad -1 \quad 5]^T$, $\eta_2(0) = [3 \quad 10 \quad 5]^T$, $b_1 = b_2 = 0.5$, $a_1 = a_2 = -0.5$. The simulation results are shown from figure 1 to figure 3.

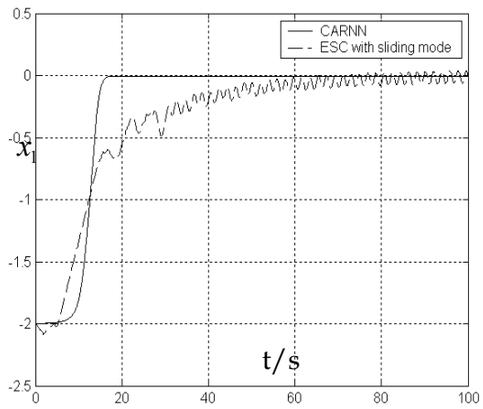


Fig. 1. The simulation result of the state x_1

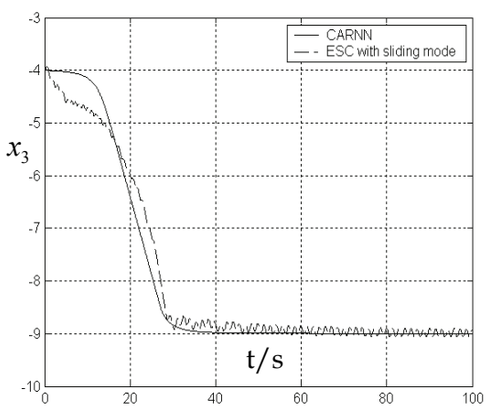


Fig. 2. The simulation result of the state x_3

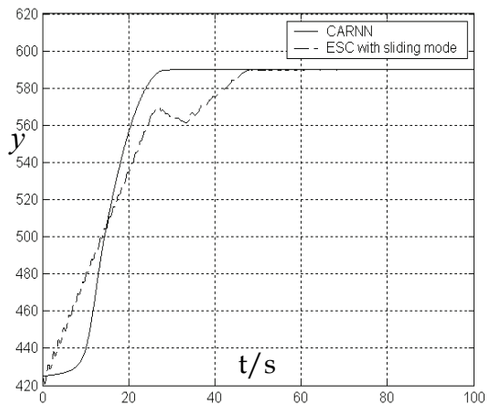


Fig. 3. The simulation result of the output y

Certainly, μ is a main factor of scaling the convergence rate of CARNN, if it is too big, the error of the output will be introduced. On the contrary, if it is too small, the convergence rate of the system will be slow. In conclusion, the values of those parameters should be verified by the system simulation.

In those simulation results, solid lines are the results applying CARNN to ESA; dash lines are the results applying ESA with sliding mode[9]. Comparing those simulation results, we know the dynamic performance of the method proposed in the paper is superior to that of ESA with sliding mode. The "chatter" of the CARNN's output doesn't exist in figure 1 and 2, which is very harmful in practice. Moreover the convergence rate of ESA with CARNN can be scaled by adjusting the chaotic annealing parameter $T(t)$.

5.2 Schaffer function simulation

In order to exhibit the capability of global searching of the proposed CARNN, the typical Schaffer function (34) is defined as the testing function[10].

$$f(x_1,x_2)=\frac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{\left(1+0.001\left(x_1^2+x_2^2\right)\right)^2}-0.5,|x_i|\le 10,i=1,2 \tag{34}$$

When $x_1=x_2=0$, the schaffer function $f(x_1,x_2)$ will obtain the global minimum $f(0,0)=-1$. There are numerous local minimums and maximums among the range of 3.14 away from the global minimum.

Now, we define $\theta_1=x_1$ and $\theta_2=x_2$. Choose $T(t)=\beta\alpha^{-\eta t}$, where $\beta=0.01$, $\alpha=e$, $\eta=3$, and apply the CARNN to search the global minimum of the function (34). The neural network parameters are given as: $\mu=35$, $\gamma=4$, $P_1(0)=P_2(0)=1$, $\kappa=0.01$, $\delta=0.001$, $\varepsilon_1=10$, $\varepsilon_2=10$, $\chi_1(0)=0.912$, $\chi_2(0)=0.551$, $\eta_1(0)=[-200\quad -20\quad 50]^T$, $\eta_2(0)=[100\quad 300\quad 50]^T$, $b_1=b_2=0.5$, $a_1=a_2=-0.5$. When the initial conditions of the function (34) are given as $x_1=-2$ and $x_2=3.5$, the simulation results are shown from figure 4 to figure 6.

When the initial conditions of the function (34) are given as $x_1(0) = -1$ and $x_2 = 9.59$, the simulation results are shown as from figure 7 to figure 9.

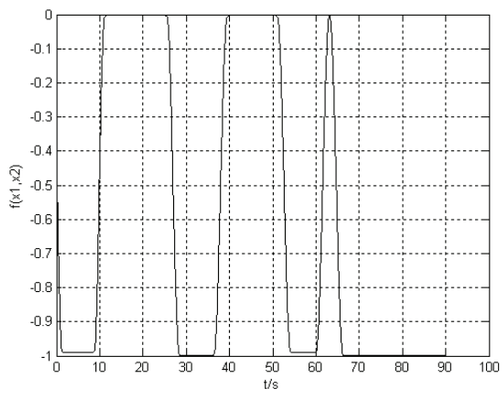


Fig. 4. The simulation result of $f(x_1, x_2)$

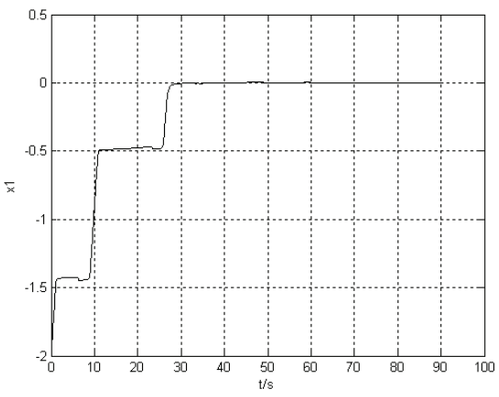


Fig. 5. The simulation result of x_1

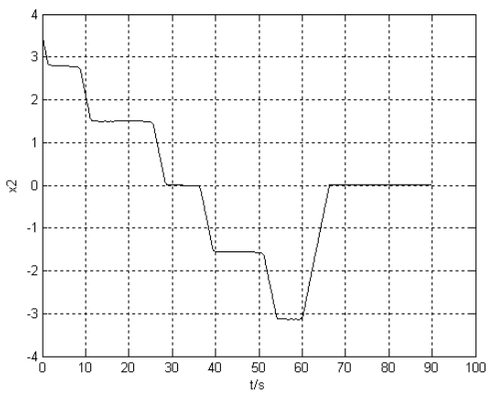


Fig. 6. The simulation result of x_2

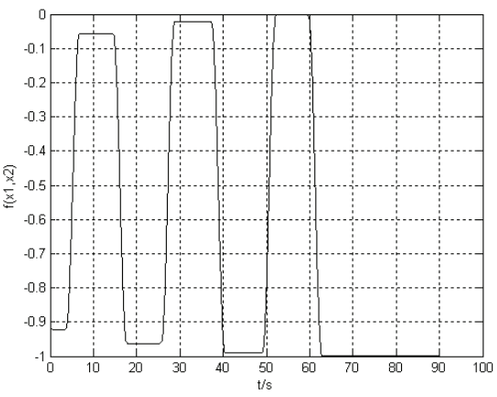


Fig. 7. The simulation result of $f(x_1, x_2)$

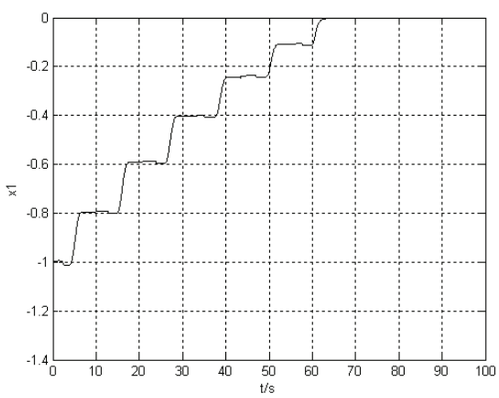


Fig. 8. The simulation result of x_1

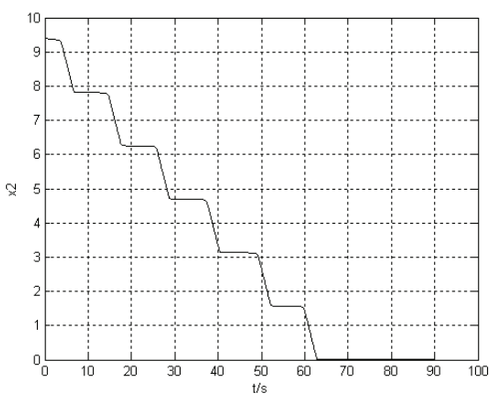


Fig. 9. The simulation result of x_2

We have accomplished a great deal of computer simulations in different initial conditions. The ESA based on the chaotic annealing recurrent neural network can find the global minimum of Schaffer function under different conditions of the simulation.

6. Referring

The method of introducing CARNN into ESA greatly improves the dynamic performance and the global searching capability of the system. Two phases of the coarse search based on chaos and the elaborate search based on ARNN guarantee that the system could fully carry out the chaos searching and find the global extremum point and accordingly converge to that point. At the same time, the disappearance of the "chatter" of the system output and the switching of the control law are beneficial to engineering applications.

7. Acknowledgements

This research was supported by the Natural Science Foundation of P.R.China (No. 60674090).

8. References

Natalia I. M. (2003). Applications of the Adaptive Extremum Seeking Control Techniques to Bioreactor Systems. A dissertation for the degree of Master of Science. Ontario: Queen's University.

Pan, Y., Ozguner, U., and Acarman, T. (2003). Stability and Performance Improvement of Extremum Seeking Control with Sliding Mode. *International Journal of Control,* Vol. 76, pp. 968-985. ISSN 0020-7179.

Drakunov, S., Ozguner, U., Dix, P., and Ashrafi, B. (1995). ABS Control Using Optimum Search via Sliding Mode., *IEEE Transactions on Control Systems Technology,* Vol. 3, No. 1, pp. 79-85. ISSN 1063-6536.

Krstic, M. (1999). Toward Faster Adaptation in Extremum Seeking Control. *Proceeding of the 38th IEEE Conference on Decision and Control,* pp. 4766-4771, ISBN 0-7803-5250-5, Phoenix, USA, Decemeber 1999.

Ying Tan, Baoyun Wang, Zhenya He. (1998). Neural Networks with Transient Chaos and Time-variant gain and Its Application to Optimization Computations. *ACTA ELECTRONICA SINICA,* Vol. 26, No. 7, pp. 123-127. ISBN 0372-2112.

Wang Ling, Zheng Dazhong. (2000). A Kind of Chaotic Neural Network Optimization Algorithm Based on Annealing Strategy. *Control Theory & Applications,* Vol. 17, No. 1, pp. 139-142. ISSN 1000-8152.

Y A Hu, B Zuo. (2005). An Annealing Recurrent Neural Network for Extremum Seeking Control. *International Journal of Information Technology,* Vol. 11, No. 6, pp. 45-52. ISSN 1305-2403.

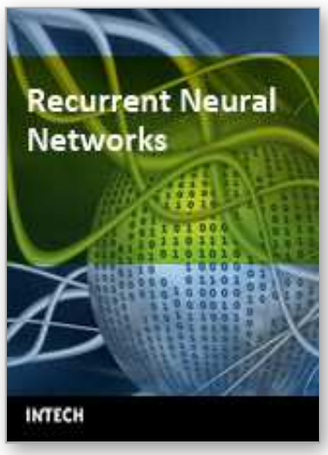
B. Zuo, and Y. A. Hu. (2004). Optimizing UAV Close Formation Flight via Extremum Seeking. *Proceedings of WCICA2004,* Vol. 4, pp. 3302-3305. ISBN 0-7803-8273-0, Hangzhou, China, June, 2004.

Yu, H., and Ozguner, U. (2002). Extremum-Seeking Control via Sliding Mode with Periodic Search Signals. *Proceeding of the 41st IEEE Conference on Decision and Control,* pp. 323-328. ISBN 0-7803-7516-5, Las Vegas, USA, December 2002.

Wang Ling. (2004). *Intelligent Optimization Algorithms with Application,* Tsinghua University Press and Springer, ISBN 7-302-04499-6, Beijing, China.

IntechOpen

IntechOpen



Recurrent Neural Networks

Edited by Xiaolin Hu and P. Balasubramaniam

ISBN 978-953-7619-08-4
Hard cover, 400 pages
Publisher InTech
Published online 01, September, 2008
Published in print edition September, 2008

The concept of neural network originated from neuroscience, and one of its primitive aims is to help us understand the principle of the central nerve system and related behaviors through mathematical modeling. The first part of the book is a collection of three contributions dedicated to this aim. The second part of the book consists of seven chapters, all of which are about system identification and control. The third part of the book is composed of Chapter 11 and Chapter 12, where two interesting RNNs are discussed, respectively.The fourth part of the book comprises four chapters focusing on optimization problems. Doing optimization in a way like the central nerve systems of advanced animals including humans is promising from some viewpoints.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yun-an Hu, Bin Zuo and Jing Li (2008). An Improved Extremum Seeking Algorithm Based on the Chaotic Annealing Recurrent Neural Network and Its Application, Recurrent Neural Networks, Xiaolin Hu and P. Balasubramaniam (Ed.), ISBN: 978-953-7619-08-4, InTech, Available from: http://www.intechopen.com/books/recurrent_neural_networks/an_improved_extremum_seeking_algorithm_based_on_the_chaotic_annealing_recurrent_neural_network_and_i

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen