Stability Analysis, Synthesis and Optimization of Radial-Basis-Function Neural-Network Based Controller for Nonlinear Systems[1]

H.K. Lam[2], *Member, IEEE*, and F.H.F. Leung, *Senior Member, IEEE*

Abstract - **This paper presents the stability analysis, synthesis, and performance optimization of a radial-basis-function neural-network based control system. Global stability conditions will be derived in terms of matrix measure. Based on the derived stability conditions, connection weights of the radial-basis-function neural-network based controller can be optimized by genetic algorithm (GA) subject to the system stability. Furthermore, the system performance will also be optimized by the GA. An application example on stabilizing an inverted pendulum will be given to illustrate the design procedure and merits of the proposed approach.**

I. Introduction

It is well known that a neural network can approximate any smooth and continuous nonlinear functions in a compact domain to an arbitrary accuracy [1]. Owing to the superior learning and generalization abilities, neural networks have been widely applied in different kinds of control problems [2]-[5]. Most of these work concentrated on the performance optimisation of neural-network based control systems, which was achieved by tuning the network parameters subject to a performance index. Still, the global system stability is one of the major issues of these control systems. In view of the highly nonlinear nature and structural complexity of the neural networks, the stability of the neural-network based control system is difficult to be analyzed.

Different neural-network based approaches with the consideration of the system stability were proposed. In [3], an adaptive neural-network based controller with variable hidden nodes was proposed. The stability of the closed-loop system is achieved by on-line parameter adaptation. The adaptive laws for parameter adjustment are derived based on the Lyapunov stability theory. In order to handle the effect of the learning error of the networks to the stability, adaptive neural-network based controllers [4] with switching control signals were proposed. These switch signals may introduce a chattering effect to the system. In [5], adaptive neural networks combined with other conventional controller were proposed. In most of these approaches, the use of the neural networks is mainly for modeling the unknown nonlinearity of the system. Another control schemes were then employed to stabilize the closed-loop system by compensating the system nonlinearity based on the learnt information of the neural networks. In summary, the system stability was achieved by adaptive and/or sliding mode control techniques in most of these approaches but not by the neural network itself. These approaches require that the network parameters are on-line changing according to some adaptive laws. This requirement will increase the computational demand, structural complexity and implementation cost of the neural-network based controller. In [6]-[7], stability conditions have been derived for a class of neural-network based control systems with a feed-forward multilayer-perceptron (MLP) neural network. The derived stability conditions were only for checking the stability of the neural-network based control system. However, the ways for finding the network parameters and optimising the system performance were ignored. These are in fact two important issues for putting the neural network controller into practice.

In this paper, the global system stability of a neural-network based control system formed by a nonlinear system and a radial-basis-function neural network (RBFNN) [8] will be investigated. The stability conditions will be derived. Subject to the derived stability conditions, an improved genetic algorithm (GA) [9] will be employed to find the solution for stability conditions and the network parameters. Furthermore, the system performance will be optimized by tuning the parameters of the activation functions of the RBFNN using the improved GA.

II. Nonlinear System and Radial-Basis-Function Neural-Network Based Controller

An RBFNN based control system consists of a nonlinear system and an RBFNN based controller.

A. Nonlinear System

The nonlinear system to be handled can be written as,

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{1}$$

where $\mathbf{x}(t) = [x_1(t) \quad x_2(t) \quad \cdots \quad x_n(t)]^T \in \Re^{n\times 1}$ is the system state vector; $\mathbf{u}(t) = [u_1(t) \quad u_2(t) \quad \cdots \quad u_m(t)]^T \in \Re^{m\times 1}$ is the input vector; $\mathbf{f}(\cdot)$ denotes a known function. It is assumed that the nonlinear system of (1) can be written as,

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{p} w_i(\mathbf{x}(t))(\mathbf{A}_i\mathbf{x}(t) + \mathbf{B}_i\mathbf{u}(t)) \tag{2}$$

where $\mathbf{A}_i \in \Re^{n\times n}$ and $\mathbf{B}_i \in \Re^{n\times m}$ are the constant input and output matrices; p is a nonzero positive integer; $w_i(\mathbf{x}(t)) \in [0 \quad 1]$ has the following property:

$$\sum_{i=1}^{p} w_i(\mathbf{x}(t)) = 1 \tag{3}$$

B. Radial-Basis-Function Neural-Network Based Controller

A traditional RBFNN [8] is shown in the Fig. 1. The input-output relationship of the RBFNN is defined as,

[1] The work described in this paper was fully supported by a grant from the The Hong Kong Polytechnic University (Project No. G-YX31). The authors are with the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

[2] email: hkl@eie.polyu.edu.hk

0-7803-8730-9/04/$20.00 ©2004 IEEE

$$y_k(t) = \sum_{j=1}^{n_h} g_{k,j} t_f\left(b_j \left\| |\mathbf{x}(t)| - \mathbf{c}_j \right\|\right), k = 1, 2, \ldots, n_{out} \quad (4)$$

where g_{kj} denotes the connection weights between the k-th output node and the j-th hidden node; $t_f(\cdot)$ denotes the activation function; b_j denotes the bias for the j-th hidden node; $\mathbf{c}_j \in \Re^{n\times 1}$ denotes the center vector; $\|\cdot\|$ denotes the l_2 norm for vectors and l_2 induced norm for matrices; $|\mathbf{x}(t)| = \left[|x_1(t)| \quad |x_2(t)| \quad \cdots \quad |x_n(t)|\right]^T$. The RBFNN based controller for the nonlinear system of (2), with $n_{out} = mn$, is defined as,

$$\mathbf{u}(t) = \frac{\begin{bmatrix} y_1(t) & y_2(t) & \cdots & y_n(t) \\ y_{n+1}(t) & y_{n+2}(t) & \cdots & y_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ y_{(m-1)n+1}(t) & y_{(m-1)n+2}(t) & \cdots & y_{mn}(t) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}}{\sum_{l=1}^{n_h} t_f\left(b_l \left\| |\mathbf{x}(t)| - \mathbf{c}_l \right\|\right)} + \mathbf{r} \quad (5)$$

where $\mathbf{r} \in \Re^{n\times 1}$ is an external input. From (4) and (5), we have,

$$\mathbf{u}(t) = \frac{\begin{bmatrix} \sum_{j=1}^{n_h} g_{1,j} t_f\left(b_j \left\| |\mathbf{x}(t)| - \mathbf{c}_j \right\|\right) & \sum_{j=1}^{n_h} g_{2,j} t_f\left(b_j \left\| |\mathbf{x}(t)| - \mathbf{c}_j \right\|\right) & \cdots & \sum_{j=1}^{n_h} g_{n,j} t_f\left(b_j \left\| |\mathbf{x}(t)| - \mathbf{c}_j \right\|\right) \\ \sum_{j=1}^{n_h} g_{n+1,j} t_f\left(b_j \left\| |\mathbf{x}(t)| - \mathbf{c}_j \right\|\right) & \sum_{j=1}^{n_h} g_{n+2,j} t_f\left(b_j \left\| |\mathbf{x}(t)| - \mathbf{c}_j \right\|\right) & \cdots & \sum_{j=1}^{n_h} g_{2n,j} t_f\left(b_j \left\| |\mathbf{x}(t)| - \mathbf{c}_j \right\|\right) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^{n_h} g_{(m-1)n+1,j} t_f\left(b_j \left\| |\mathbf{x}(t)| - \mathbf{c}_j \right\|\right) & \sum_{j=1}^{n_h} g_{(m-1)n+2,j} t_f\left(b_j \left\| |\mathbf{x}(t)| - \mathbf{c}_j \right\|\right) & \cdots & \sum_{j=1}^{n_h} g_{mn,j} t_f\left(b_j \left\| |\mathbf{x}(t)| - \mathbf{c}_j \right\|\right) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}}{\sum_{l=1}^{n_h} t_f\left(b_l \left\| |\mathbf{x}(t)| - \mathbf{c}_l \right\|\right)} + \mathbf{r}$$

$$= \frac{\left(\sum_{j=1}^{n_h} t_f\left(b_j \left\| |\mathbf{x}(t) - \mathbf{c}_j \right\|\right)\right) \begin{bmatrix} g_{1,j} & g_{2,j} & \cdots & g_{n,j} \\ g_{n+1,j} & g_{n+2,j} & \cdots & g_{2n,j} \\ \vdots & \vdots & \ddots & \vdots \\ g_{(m-1)n+1,j} & g_{(m-1)n+2,j} & \cdots & g_{mn,j} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}}{\sum_{l=1}^{n_h} t_f\left(b_l \left\| |\mathbf{x}(t)| - \mathbf{c}_l \right\|\right)} + \mathbf{r} = \sum_{j=1}^{n_h} m_j(\mathbf{x}(t)) \mathbf{G}_j \mathbf{x}(t) + \mathbf{r} \quad (6)$$

where

$$\mathbf{G}_j = \begin{bmatrix} g_{1,j} & g_{2,j} & \cdots & g_{n,j} \\ g_{n+1,j} & g_{n+2,j} & \cdots & g_{2n,j} \\ \vdots & \vdots & \ddots & \vdots \\ g_{(m-1)n+1,j} & g_{(m-1)n+2,j} & \cdots & g_{mn,j} \end{bmatrix} \quad (7)$$

$$m_j(\mathbf{x}(t)) = \frac{t_f\left(b_j \left\| |\mathbf{x}(t)| - \mathbf{c}_j \right\|\right)}{\sum_{l=1}^{n_h} t_f\left(b_l \left\| |\mathbf{x}(t)| - \mathbf{c}_l \right\|\right)} \in [0 \quad 1] \quad (8)$$

which satisfies the property of $w_i(\mathbf{x}(t))$ in (3). It is assumed that $\mathbf{c}_l$ is chosen such that $\sum_{l=1}^{n_h} t_f\left(b_l \left\| |\mathbf{x}(t)| - \mathbf{c}_l \right\|\right) \neq 0$ at any time.

C. Radial-Basis-Function Neural-Network Based Control Systems

An RBFNN control system is formed by connecting the nonlinear system of (2) and an RBFNN controller of (6) in closed-loop. From (2) and (6), writing $w_i(\mathbf{x}(t))$ as w_i and $m_k(\mathbf{x}(t))$ as m_k, we have the closed-loop system in the following form:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{p} w_i \left(\mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \left(\sum_{j=1}^{n_h} m_j \mathbf{G}_j \mathbf{x}(t) + \mathbf{r} \right) \right) \quad (9)$$

Because $\sum_{i=1}^{p} w_i = \sum_{j=1}^{n_h} m_j = \sum_{i=1}^{p} \sum_{j=1}^{n_h} w_i m_j = 1$, (9) can be re-written as,

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{p} \sum_{j=1}^{n_h} w_i m_j \left(\left(\mathbf{A}_i + \mathbf{B}_i \mathbf{G}_j \right) \mathbf{x}(t) + \mathbf{B}_i \mathbf{r} \right)$$

$$= \sum_{i=1}^{p} \sum_{j=1}^{n_h} w_i m_j \left(\mathbf{H}_{ij} \mathbf{x}(t) + \mathbf{B}_i \mathbf{r} \right) \quad (10)$$

$$\mathbf{H}_{ij} = \mathbf{A}_i + \mathbf{B}_i \mathbf{G}_j \quad (11)$$

The system stability of the closed-loop system of (10) will be analyzed in the following section.

III. Stability Analysis of Radial-Basis-Function Neural-Network Based Control Systems

In the following, the stability of the closed-loop system of (10) will be analysed. Consider the Taylor series,

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \dot{\mathbf{x}}(t)\Delta t + \mathbf{o}(\Delta t) \quad (12)$$

where $\mathbf{o}(\Delta t) = -\mathbf{x}(t) - \dot{\mathbf{x}}(t)\Delta t + \mathbf{x}(t + \Delta t)$ is the error term and $\Delta t > 0$,

$$\lim_{\Delta t \to 0^+} \frac{\|\mathbf{o}(\Delta t)\|}{\Delta t} = 0 \quad (13)$$

From (10) and (12), and multiplying a transformation matrix $\mathbf{T} \in \Re^{n\times n}$ of rank n to both sides, we have

$$\mathbf{T}\mathbf{x}(t + \Delta t) = \mathbf{T}\mathbf{x}(t) + \sum_{i=1}^{p} \sum_{j=1}^{n_h} w_i m_j \mathbf{T} \left(\mathbf{H}_{ij} \mathbf{x}(t) + \mathbf{B}_i \mathbf{r} \right) \Delta t + \mathbf{T}\mathbf{o}(\Delta t)$$

$$= \left(\mathbf{I} + \sum_{i=1}^{p} \sum_{j=1}^{n_h} w_i m_j \mathbf{T} \mathbf{H}_{ij} \mathbf{T}^{-1} \Delta t \right) \mathbf{T}\mathbf{x}(t) + \sum_{i=1}^{p} \sum_{j=1}^{n_h} w_i m_j \mathbf{T} \mathbf{B}_i \mathbf{r} \Delta t + \mathbf{T}\mathbf{o}(\Delta t)$$

The reason for introducing $\mathbf{T}$ will be given at the end of this section. Taking norm on both sides of the above equation,

$$\|\mathbf{Tx}(t+\Delta t)\| \le \left\|\sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \left(\mathbf{I}+\mathbf{TH}_{ij}\mathbf{T}^{-1}\Delta t\right)\right\|\|\mathbf{Tx}(t)\| + \left\|\sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \mathbf{TB}_i\mathbf{r}\Delta t\right\| + \|\mathbf{To}(\Delta t)\| \quad (14)$$

From (14),

$$\|\mathbf{Tx}(t+\Delta t)\| \le \sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \left\|\mathbf{I}+\mathbf{TH}_{ij}\mathbf{T}^{-1}\Delta t\right\|\|\mathbf{Tx}(t)\| + \left\|\sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \mathbf{TB}_i\mathbf{r}\Delta t\right\| + \|\mathbf{To}(\Delta t)\|$$

$$\Rightarrow \|\mathbf{Tx}(t+\Delta t)\| - \|\mathbf{Tx}(t)\| \le \sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \left(\left\|\mathbf{I}+\mathbf{TH}_{ij}\mathbf{T}^{-1}\Delta t\right\|-1\right)\|\mathbf{Tx}(t)\| + \left\|\sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \mathbf{TB}_i\mathbf{r}\Delta t\right\| + \|\mathbf{To}(\Delta t)\|$$

$$\Rightarrow \lim_{\Delta t\to 0^+}\frac{\|\mathbf{Tx}(t+\Delta t)\| - \|\mathbf{Tx}(t)\|}{\Delta t} \le \lim_{\Delta t\to 0^+}\left[\sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \left(\left\|\mathbf{I}+\mathbf{TH}_{ij}\mathbf{T}^{-1}\Delta t\right\|-1\right)\|\mathbf{Tx}(t)\| + \left\|\sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \mathbf{TB}_i\mathbf{r}\Delta t\right\| + \|\mathbf{To}(\Delta t)\|\right]/\Delta t \quad (15)$$

From (13) and (15),

$$\frac{\mathrm{d}\|\mathbf{Tx}(t)\|}{\mathrm{d}t} \le \lim_{\Delta t\to 0^+}\frac{\sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \left(\left\|\mathbf{I}+\mathbf{TH}_{ij}\mathbf{T}^{-1}\Delta t\right\|-1\right)}{\Delta t}\|\mathbf{Tx}(t)\| + \left\|\sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \mathbf{TB}_i\mathbf{r}\right\|$$

$$\le \sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \mu\left[\mathbf{TH}_{ij}\mathbf{T}^{-1}\right]\|\mathbf{Tx}(t)\| + \left\|\sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \mathbf{TB}_i\mathbf{r}\right\| \quad (16)$$

where,

$$\mu\left[\mathbf{TH}_{ij}\mathbf{T}^{-1}\right] = \lim_{\Delta t\to 0^+}\frac{\left\|\mathbf{I}+\mathbf{TH}_{ij}\mathbf{T}^{-1}\Delta t\right\|-1}{\Delta t} = \lambda_{\max}\left(\frac{\mathbf{TH}_{ij}\mathbf{T}^{-1}+(\mathbf{TH}_{ij}\mathbf{T}^{-1})^*}{2}\right) \quad (17)$$

is the corresponding matrix measure [10] of the induced matrix norm of $\|\mathbf{TH}_{ij}\mathbf{T}^{-1}\|$ (or the logarithmic derivative of $\|\mathbf{TH}_{ij}\mathbf{T}^{-1}\|$); $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue, * denotes the conjugate transpose. From (16),

$$\frac{\mathrm{d}\|\mathbf{Tx}(t)\|}{\mathrm{d}t} \le \sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \mu\left[\mathbf{TH}_{ij}\mathbf{T}^{-1}\right]\|\mathbf{Tx}(t)\| + \left\|\sum_{i=1}^{p} w_i \mathbf{TB}_i\mathbf{r}\right\| \quad (18)$$

Let $\mu\left[\mathbf{TH}_{ij}\mathbf{T}^{-1}\right]$ be designed to satisfy the following inequalities:

$$\mu\left[\mathbf{TH}_{ij}\mathbf{T}^{-1}\right] \le -\varepsilon \text{ for all } i \text{ and } j. \quad (19)$$

where ε is a designed nonzero positive constant, it can be proved that (18) implies a stable system of (10). Before conducting the proof, consider the following inequality obtained from (18) and (19):

$$\frac{\mathrm{d}\|\mathbf{Tx}(t)\|}{\mathrm{d}t} \le -\sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i m_j \varepsilon\|\mathbf{Tx}(t)\| + \sum_{i=1}^{p} w_i\|\mathbf{TB}_i\mathbf{r}\|$$

$$= -\varepsilon\|\mathbf{Tx}(t)\| + \sum_{i=1}^{p} w_i\|\mathbf{TB}_i\mathbf{r}\|$$

$$\Rightarrow \left(\frac{\mathrm{d}\|\mathbf{Tx}(t)\|}{\mathrm{d}t} + \varepsilon\|\mathbf{Tx}(t)\|\right)e^{\varepsilon(t-t_o)} \le \sum_{i=1}^{p} w_i\|\mathbf{TB}_i\mathbf{r}\|e^{\varepsilon(t-t_o)}$$

$$\Rightarrow \frac{\mathrm{d}}{\mathrm{d}t}\left(\|\mathbf{Tx}(t)\|e^{\varepsilon(t-t_o)}\right) \le \sum_{i=1}^{p} w_i\|\mathbf{TB}_i\mathbf{r}\|e^{\varepsilon(t-t_o)} \quad (20)$$

where $t_o < t$ is an arbitrary initial time. Based on (20), there are two cases to investigate the system behavior: $\mathbf{r} = \mathbf{0}$ and $\mathbf{r} \neq \mathbf{0}$. For the former case, it can be shown that if the condition of (19) is satisfied, the closed-loop system of (10) is exponentially stable, and $\|\mathbf{x}(t)\| \to 0$ as $t \to \infty$.

Proof. For $\mathbf{r} = \mathbf{0}$, from (20),

$$\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{Tx}(t)\|e^{\varepsilon(t-t_o)} \le 0 \quad \Rightarrow \|\mathbf{Tx}(t)\|e^{\varepsilon(t-t_o)} \le \|\mathbf{Tx}(t_o)\|$$

$$\Rightarrow \|\mathbf{Tx}(t)\| \le \|\mathbf{Tx}(t_o)\|e^{-\varepsilon(t-t_o)} \quad (21)$$

Since ε is a positive value, $\|\mathbf{Tx}(t)\| \to 0$ as $t \to \infty$. Also,

$$\sigma_{\min}(\mathbf{T}^{\mathrm{T}}\mathbf{T})\|\mathbf{x}(t)\|^2 \le \|\mathbf{Tx}(t)\|^2 = \mathbf{x}(t)^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\mathbf{Tx}(t) \le \sigma_{\max}(\mathbf{T}^{\mathrm{T}}\mathbf{T})\|\mathbf{x}(t)\|^2 \quad (22)$$

where $\sigma_{\max}(\mathbf{T}^{\mathrm{T}}\mathbf{T})$ and $\sigma_{\min}(\mathbf{T}^{\mathrm{T}}\mathbf{T})$ denote the maximum and minimum singular values of $\mathbf{T}^{\mathrm{T}}\mathbf{T}$ respectively. As $\mathbf{T}^{\mathrm{T}}\mathbf{T}$ is symmetric positive definite ($\mathbf{T}$ has rank n), from (22), $\|\mathbf{Tx}(t)\| \to 0$ only when $\|\mathbf{x}(t)\| \to 0$. **QED**

For the latter case of $\mathbf{r} \neq \mathbf{0}$, the closed-loop system of (10) is input-to-state stable, i.e. the system states are bounded if the condition of (19) is satisfied and $\mathbf{r}$ is bounded.

Proof. For $\mathbf{r} \neq \mathbf{0}$, from (20),

$$\|\mathbf{Tx}(t)\|e^{\varepsilon(t-t_o)} \le \|\mathbf{Tx}(t_o)\| + \int_{t_o}^{t}\sum_{i=1}^{p} w_i\|\mathbf{TB}_i\mathbf{r}\|e^{\varepsilon(\tau-t_o)}\mathrm{d}\tau$$

$$\Rightarrow \|\mathbf{Tx}(t)\|e^{\varepsilon(t-t_o)} \le \|\mathbf{Tx}(t_o)\| + \|\mathbf{TB}_i\mathbf{r}\|_{\max}\int_{t_o}^{t} e^{\varepsilon(\tau-t_o)}\mathrm{d}\tau$$

where $\max_i\|\mathbf{TB}_i\mathbf{r}\|_{\max} \ge \|\mathbf{TB}_i\mathbf{r}\|$. Then

$$\|\mathbf{Tx}(t)\|e^{\varepsilon(t-t_o)} \le \|\mathbf{Tx}(t_o)\| + \frac{\|\mathbf{TB}_i\mathbf{r}\|_{\max}}{\varepsilon}(e^{\varepsilon(t-t_o)}-1)$$

$$\Rightarrow \|\mathbf{Tx}(t)\| \le \|\mathbf{Tx}(t_o)\|e^{-\varepsilon(t-t_o)} + \frac{\|\mathbf{TB}_i\mathbf{r}\|_{\max}}{\varepsilon}(1-e^{-\varepsilon(t-t_o)}) \quad (23)$$

Since the right hand side of (23) is finite if $\mathbf{r}$ is bounded, the system states are also bounded. **QED**

The stability conditions of the closed-loop system are summarized by the following lemma:

Lemma 1. *The RBFNN based control system as given by* (10) *is exponentially stable for* $\mathbf{r} = \mathbf{0}$ *or input-to-state stable for* $\mathbf{r} \neq \mathbf{0}$ *if* $\mathbf{TH}_{ij}\mathbf{T}^{-1}$ *is designed such that,*

$$\mu\left[\mathbf{TH}_{ij}\mathbf{T}^{-1}\right] \le -\varepsilon \text{ for all } i \text{ and } j$$

where ε *is a nonzero positive constant scalar.*

It should be noted that with the use of a suitable transformation matrix $\mathbf{T}$, any Hurwitz matrix having a

positive or zero matrix measure can be transformed into another matrix having a negative matrix measure (see (19)). The stability conditions derived can then be applied. The problem left is how to find such a matrix $\mathbf{T}$ for a given system. This will be discussed later. From the above derivation and Lemma 1, we can see the system stability is not affected by 1) $w_i(\mathbf{x}(t))$ of the nonlinear plant and 2) the activation function $t_j(\cdot)$ of the RBFNN based controller. These properties 1) and 2) help the robustness of the closed-loop system. With the property 1), if we incorporate parameter uncertainties into $w_i(\mathbf{x}(t))$, the proposed RBFNN based controller is able to stabilize the closed-loop system subject to parameter uncertainties if Lemma 1 satisfied. With the property 2), the parameters of the RBFNN activation functions can be tuned to obtain the optimal system performance without affecting the system stability.

IV. Design and Optimization of Radial-Basis-Function Neural-Network Based Controller

In this section, the problems of solving the stability conditions derived in the previous section, obtaining the network parameters of the RBFNN, and optimizing the system performance are tackled using the improved GA [9].

A. Solving the Stability Conditions and Obtaining the Feedback Gains

From Lemma 1, the RBFNN based control system is stable if the following conditions hold:

$$\mu\left[\mathbf{T}\left(\mathbf{A}_i+\mathbf{B}_i\mathbf{G}_j\right)\mathbf{T}^{-1}\right] \le -\varepsilon,\ i = 1, 2, \ldots, p;\ j = 1, 2, \ldots, n_h \tag{24}$$

We therefore have to find $\mathbf{T}=\begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1n} \\ T_{21} & T_{22} & \cdots & T_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ T_{n1} & T_{n2} & \cdots & T_{nn} \end{bmatrix}$ and $\mathbf{G}_j=\begin{bmatrix} g_{1,j} & g_{2,j} & \cdots & g_{n,j} \\ g_{n+1,j} & g_{n+2,j} & \cdots & g_{2n,j} \\ \vdots & \vdots & \ddots & \vdots \\ g_{(m-1)n+1,j} & g_{(m-1)n+2,j} & \cdots & g_{mn,j} \end{bmatrix}$ such that the conditions of (24) are satisfied. Define a fitness function as follows:

$$fitness = \sum_{i=1}^{p}\sum_{j=1}^{n_h} n_{ij}\,\mu\left[\mathbf{T}\left(\mathbf{A}_i+\mathbf{B}_i\mathbf{G}_j\right)\mathbf{T}^{-1}\right] \tag{25}$$

where $n_{ij} > 0$, $i = 1, 2, \ldots, p$; $j = 1, 2, \ldots, n_h$, are constant scalar. The problems of finding $\mathbf{T}$ and $\mathbf{G}_j$ can be formulated into a minimization problem. The aim is to minimize the fitness function of (25) by tuning $\mathbf{T}$ and $\mathbf{G}_j$ using the improved GA. As $\mathbf{T}$ and $\mathbf{G}_j$ are the variables of the fitness function of (25), they will be used to form the genes of the chromosomes. The finding of the solution to this minimization problem, however, does not immediately imply that the conditions of (24) are satisfied. Hence, different n_{ij}, $i = 1, 2, \ldots, p$; $j = 1, 2, \ldots, n_h$, may need to weight the conditions of (24) in order to change the significance of different terms on the right hand side of (25). For instance, the value of one of the terms in (25) is very large, which returns a very large fitness value. Under this case, the conditions of (24) are not satisfied. A large value of n_{ij} corresponding to that term can be used to increase the tuning effect to that term in the fitness function. This may help the GA process to find a solution that satisfies the conditions of (24) during the minimizing process.

B. Optimizing the System Performance

After $\mathbf{T}$ and $\mathbf{G}_j$ have been determined, what follows is to determine the parameters of the activation functions of the RBFNN based controller using the improved GA [9] such that the performance of the RBFNN control system is optimal subject to a defined performance index. The dynamics of the RBFNN control system are restated as follows,

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{p}\sum_{j=1}^{n_h} w_i(\mathbf{x}(t))m_j(\mathbf{x}(t),\mathbf{z}_j)\left(\mathbf{H}_{ij}\mathbf{x}(t)+\mathbf{B}_i\mathbf{r}\right) \tag{26}$$

where $\mathbf{z}_j$ is the parameter vector governing the j-th activation function of the RBFNN based controller, e.g. the values of b_j and $\mathbf{c}_j$. A performance index J which measures the system performance is defined as follows,

$$J = \int \mathbf{x}(t)^{\mathrm{T}}\mathbf{W}_x\mathbf{x}(t)+\mathbf{u}(t)^{\mathrm{T}}\mathbf{R}_u\mathbf{u}(t)\mathrm{d}t \tag{27}$$

where $\mathbf{W}_x \in \Re^{n\times n}$ and $\mathbf{R}_u \in \Re^{m\times m}$ are both constant semi-positive or positive definite matrices. This is the performance index used in the conventional optimal control [10]. The optimization problem formulated here is handled by the improved GA. $\mathbf{z}_j, j = 1, 2, \ldots, n_h$, will be used to form the genes of the chromosomes for the GA process. The procedure to obtain the RBFNN based controller using the improved GA can be summarized into the following steps:

I) Obtain the mathematical model of the nonlinear plant. Convert the mathematical model into the form of the plant model of (2).

II) Determine the number of hidden nodes for the RBFNN controller. Solve $\mathbf{T}$ and $\mathbf{G}_j$ with the fitness function defined in (25) with $n_{ij} = 1$ initially, $i = 1, 2, \ldots, p$; $j = 1, 2, \ldots, n_h$, using the improved GA. If $\mathbf{T}$ and $\mathbf{G}_j$ cannot be found, adjust n_{ij} accordingly.

III) Determine the activation functions of the RBFNN based controller. Obtain the parameters of the activation functions by optimizing the system performance with respect to the performance index of (27) using the improved GA.

V. Application Example

An application example on stabilising a cart-pole typed inverted pendulum is given in this section. An RBFNN based controller will be used to control the plant.

Step I) The dynamic equation of the cart-pole typed inverted pendulum is given by,

$$\ddot{\theta}(t) = \frac{g\sin(\theta(t)) - aml\dot{\theta}(t)^2\sin(2\theta(t))/2 - a\cos(\theta(t))u(t)}{4l/3 - aml\cos^2(\theta(t))} \tag{28}$$

where θ is the angular displacement of the pendulum, $g = 9.8\text{m/s}^2$ is the acceleration due to gravity, $m = 2\text{kg}$ is the mass of the pendulum, $M = 8\text{kg}$ is the mass of the cart, $a = 1/(m + M)$, $2l = 1\text{m}$ is the length of the pendulum, and u is the force

applied to the cart. The objective of this application example is to design a controller to close the feedback loop of (28) such that $\theta = 0$ at steady state. The nonlinear plant can be written in the following form,

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{4} w_i \left(\mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i u(t) \right) \tag{29}$$

where $\mathbf{x}(t) = [x_1(t) \quad x_2(t)]^{\mathrm{T}} = [\theta(t) \quad \dot{\theta}(t)]^{\mathrm{T}}$

$\theta(t) \in [\theta_{\min} \quad \theta_{\max}] = \left[-\frac{22\pi}{45} \quad \frac{22\pi}{45} \right]$ and

$\dot{\theta}(t) \in [\dot{\theta}_{\min} \quad \dot{\theta}_{\max}] = [-5 \quad 5]$;

$f_1(\mathbf{x}(t)) = \dfrac{g - aml x_2(t)^2 \cos(x_1(t))}{4l/3 - aml \cos^2(x_1(t))} \left(\dfrac{\sin(x_1(t))}{x_1(t)} \right)$ and

$f_2(\mathbf{x}(t)) = -\dfrac{a \cos(x_1(t))}{4l/3 - aml \cos^2(x_1(t))}$; $\mathbf{A}_1 = \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ f_{1_{\min}} & 0 \end{bmatrix}$

and $\mathbf{A}_3 = \mathbf{A}_4 = \begin{bmatrix} 0 & 1 \\ f_{1_{\max}} & 0 \end{bmatrix}$; $\mathbf{B}_1 = \mathbf{B}_3 = \begin{bmatrix} 0 \\ f_{2_{\min}} \end{bmatrix}$ and

$\mathbf{B}_2 = \mathbf{B}_4 = \begin{bmatrix} 0 \\ f_{2_{\max}} \end{bmatrix}$; $f_{1_{\min}} = 9$ and $f_{1_{\max}} = 18$,

$f_{2_{\min}} = -0.1765$ and $f_{2_{\max}} = -0.0052$;

$w_i = \dfrac{\mu_{M_1^i}(f_1(\mathbf{x}(t))) \times \mu_{M_2^i}(f_2(\mathbf{x}(t)))}{\sum_{l=1}^{4} \left(\mu_{M_1^l}(f_1(\mathbf{x}(t))) \times \mu_{M_2^l}(f_2(\mathbf{x}(t))) \right)}$;

$\mu_{M_1^\beta}(f_1(\mathbf{x}(t))) = \dfrac{-f_1(\mathbf{x}(t)) + f_{1_{\max}}}{f_{1_{\max}} - f_{1_{\min}}}$ for $\beta = 1, 2$;

$\mu_{M_1^\delta}(f_1(\mathbf{x}(t))) = 1 - \mu_{M_1^1}(f_1(\mathbf{x}(t)))$ for $\delta = 3, 4$;

$\mu_{M_2^\kappa}(f_2(\mathbf{x}(t))) = \dfrac{-f_2(\mathbf{x}(t)) + f_{2_{\max}}}{f_{2_{\max}} - f_{2_{\min}}}$ for $\kappa = 1, 3$

and $\mu_{M_2^\phi}(f_2(\mathbf{x}(t))) = 1 - \mu_{M_2^3}(f_2(\mathbf{x}(t)))$ for $\phi = 2, 4$.

Step II) The following RBFNN based controller with $n_h = 4$ is designed for the plant of (29).

$$u(t) = \sum_{j=1}^{4} m_j(\mathbf{x}(t)) \mathbf{G}_j \mathbf{x}(t) \tag{30}$$

To guarantee the closed-loop system stability and obtain the connection weights $\mathbf{G}_j$ of the RBFNN based controller of (30), we have to solve $\mathbf{T}$ and $\mathbf{G}_j$, $j = 1, 2, 3, 4$, using the improved GA with the following fitness function:

$$fitness = \sum_{i=1}^{4} \sum_{j=1}^{4} n_{ij} \mu \left[\mathbf{T} \left(\mathbf{A}_i + \mathbf{B}_i \mathbf{G}_j \right) \mathbf{T}^{-1} \right] \tag{31}$$

The minimum and maximum values of each element of $\mathbf{T}$ are chosen to be -1 and 1 respectively. The minimum and maximum values of each element of $\mathbf{G}_1$ to $\mathbf{G}_4$ are chosen to be 0 and 5500 respectively. All n_{ij} are chosen to be 1. The initial values of the elements of $\mathbf{T}$ and $\mathbf{G}_j$ are randomly generated. The control parameters of the improved GA [9], namely the weight w and the probability of acceptance p_a are chosen to be 0.5 and 0.1 respectively. The population size and the number of iteration are chosen to be 10 and 2000 respectively. After applying the improved GA process, we obtain

$\mathbf{T} = \begin{bmatrix} 0.9668 & 0.1015 \\ -0.5053 & -0.2213 \end{bmatrix}$ and

$\mathbf{G}_1 = [3499.2662 \quad 1019.1868]$,

$\mathbf{G}_2 = [3491.1217 \quad 982.8978]$, $\mathbf{G}_3 = [090.8724 \quad 982.9060]$ and $\mathbf{G}_4 = [5383.9705 \quad 1543.4943]$. The stability analysis result is tabulated in Table I. It can be seen that the values of $\mu[\mathbf{T}(\mathbf{A}_i + \mathbf{B}_i \mathbf{G}_j)\mathbf{T}^{-1}]$, $i = 1, 2, 3, 4$; $j = 1, 2, 3, 4$, are all negative. According to Lemma 1, the RBFNN based control system is guaranteed to be exponentially stable. Based on this table, we can arbitrarily choose ε to be 0.01.

Step III) The optimal performance of the RBFNN based control system could be obtained by tuning the parameters of the activation functions. The activation function $t_f(\cdot)$ used by the RBFNN is the Gaussian function,

$$t_f\left(b_k \left\| \mathbf{x}(t) - \mathbf{c}_k \right\| \right) = e^{\frac{-\left(b_k \left\| \mathbf{x}(t) - \mathbf{c}_k \right\| \right)^2}{2\sigma_k^2}} \tag{32}$$

where σ_k is a nonzero positive scalar governing the standard deviation of the Gaussian function of (32). In this example, the tunable parameters are b_k, $\mathbf{c}_k$ and σ_k, $k = 1, 2, 3, 4$, which form the chromosomes of the GA process. The minimum and maximum values of b_k are chosen to be -10 and 10 respectively, those of the elements of $\mathbf{c}_k$ are chosen to be $-\pi$ and π respectively and those of σ_i are chosen to be 0.0001 and 1 respectively. The control parameters of w and p_a of the improved GA [9] are chosen to be 0.5 and 0.1 respectively. The population size and the number of iteration are chosen to be 10 and 500 respectively. The performance index is chosen as follows,

$$J = \int_0^\infty \mathbf{x}(t)^{\mathrm{T}} \mathbf{W}_x \mathbf{x}(t) + \mathbf{u}(t)^{\mathrm{T}} \mathbf{R}_u \mathbf{u}(t) dt \tag{33}$$

where $\mathbf{W}_x = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}$ and $\mathbf{R}_u = 10^{-6}$. Table II lists the values of the parameters of the activation functions of the RBFNN controller initially and after the GA process. The fitness values before and after the GA process are 0.0208 and 0.0182 respectively. It is about 12.5% improvement in terms of fitness value. Fig. 2 shows the responses of the system states before (dotted lines) and after (solid lines) the GA process under the initial conditions of $\mathbf{x}(0) = \left[\frac{22\pi}{45} \quad 0 \right]^{\mathrm{T}}$, $\mathbf{x}(0) = \left[\frac{11\pi}{45} \quad 0 \right]^{\mathrm{T}}$, $\mathbf{x}(0) = \left[-\frac{11\pi}{45} \quad 0 \right]^{\mathrm{T}}$ and $\mathbf{x}(0) = \left[-\frac{22\pi}{45} \quad 0 \right]^{\mathrm{T}}$. It can be seen from the simulation results that the responses after optimization are better in terms of settling time. In this paper, the derived stability conditions form the basis for designing an RBFNN based controller. From this application example, it can be seen that finding the connection weights of the RBFNN based controller is mainly for the system stability, which will not be affected by the parameters of the activation functions of the RBFNN based controller. To improve the system performance, we can further tune the parameters of the activation functions of the

RBFNN based controller.

VI. Conclusion

The stability of radial-basis-function neural-network based control systems has been analyzed. The stability conditions have been derived for this class of nonlinear systems. The design of the network parameters and the optimization of the closed-loop system performance have been tuned by GA. An application example on stabilizing an inverted pendulum by using the proposed approach has been given.

References

[1] B. Widrow and M.A. Lehr, "30 years of adaptive neural networks: Perceptron, madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415-1442, Sept. 1990.

[2] D. Nguyen and B. Widrom, "The truck backer-upper: An example of self-learning in neural networks," in *Proc. Int. Joint Conf. Neural Networks (IJCNN-89)*, vol. 2, June 1989, pp. 357-363.

[3] G.P. Liu, V. Kadirkamanathan, and S.A. Billings, "Variable neural networks for adaptive control of nonlinear systems," *IEEE Transactions on System, Man, and Cybernetic – Part C: Applications and Reviews*, vol. 29, no.1, pp. 34-43, Feb. 1999.

[4] H.D. Patiño, R. Carelli, and B.R. Kuchen, "Neural network for advanced control of robot manipulators," *IEEE Transactions on Neural Networks*, vol. 13, no.2, pp. 343-354, March 2002.

[5] S. Lin and A.A. Goldenberg, "Neural-network control of mobile manipulators," *IEEE Transactions on Neural Networks*, vol. 12, no. 5, pp. 1121-1133, Sept. 2001.

[6] K. Tanaka, "An approach to stability criteria of neural-network control systems," *IEEE Transactions on Neural Network*, vol. 7, no. 3, pp. 629-642, May 1996.

[7] J.D. Hwang and F.H. Hsiao, "Stability analysis of neural network interconnected systems," vol. 14, no. 1, pp. 201-208, Jan. 2003.

[8] J.M. Zurada, *Introduction to Artificial Neural Systems*. West Info Access, 1992.

[9] F.H.F. Leung, H.K. Lam, S.H. Ling and P.K.S. Tam, "Tuning of the structure and parameters of neural network using an improved genetic algorithm," *IEEE Trans. Neural Networks*, vol. 14, no. 1, pp. 79-88, Jan. 2003.

[10] M. Vidyasagar, *Nonlinear Systems Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1993.

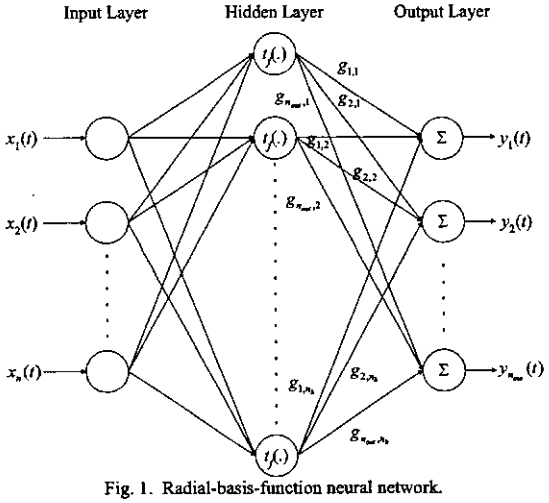

Fig. 1. Radial-basis-function neural network.

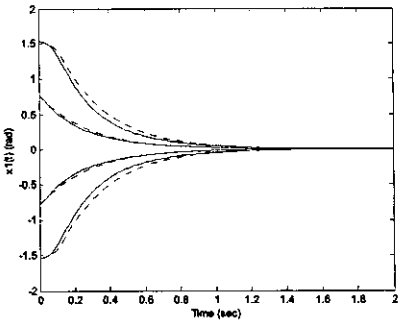

Fig. 2(a). Responses of $x_1(t)$.

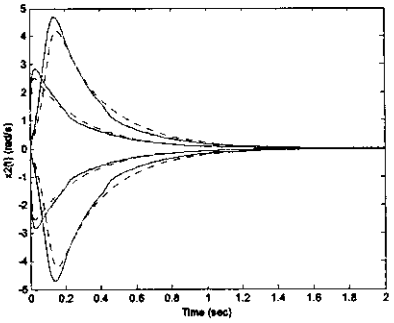

Fig. 2(b). Responses of $x_2(t)$.

Fig. 2. System responses of the RBFNN based control system before (dotted lines) and after (solid lines) the GA process with $\mathbf{W}_z = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}$ and $\mathbf{R}_w = 0$.

i,j	$\mu[\mathbf{TH}_{ij}\mathbf{T}^{-1}]$	i,j	$\mu[\mathbf{TH}_{ij}\mathbf{T}^{-1}]$
1, 1	–6.8721	3, 1	–6.6792
1, 2	–7.0811	3, 2	–7.033
1, 3	–2.8489	3, 3	–3.545
1, 4	–6.9972	3, 4	–6.8936
2, 1	–2.9245	4, 1	–0.0936
2, 2	–2.7369	4, 2	–0.05049
2, 3	–3.1883	4, 3	–1.0857
2, 4	–6.1871	4, 4	–3.0906

Table I. Stability analysis results of the inverted pendulum system with the RBFNN based controller.

Parameters of Gaussian functions	Initial values before GA tuning	Final values after GA tuning with $\mathbf{W}_z = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}$ and $\mathbf{R}_w = 0$
b_1	0.5	8.4977
b_2	0.5	7.3491
b_3	0.5	9.8939
b_4	0.5	9.4424
$\mathbf{c}_1$	[0 0]	[–1.7054 2.4602]
$\mathbf{c}_2$	[0 0]	[–1.1483 –2.0752]
$\mathbf{c}_3$	[0 0]	[–0.9944 2.3857]
$\mathbf{c}_4$	[0 0]	[0.08103 –2.3546]
σ_1	0.5	0.30245
σ_2	0.5	0.29947
σ_3	0.5	0.40783
σ_4	0.5	0.86255

Table II. Values of the parameters of the activation functions of the RBFNN based controller before and after the GA process.