PERGAMON

Mechatronics 8 (1998) 887–903

MECHATRONICS

Adaptive neural network feedback control of a passive line-of-sight stabilization system

T. H. Lee*, S. S. Ge, C. P. Wong

Centre for Intelligent Control, Department of Electrical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260

Received 5 February 1997; accepted 28 April 1998

Abstract

An adaptive neural network full-state feedback controller has been designed and applied to the passive line-of-sight (LOS) stabilization system. Model reference adaptive control (MRAC) is well established for linear systems. However, this method cannot be utilized directly since the LOS system is nonlinear in nature. Utilizing the universal approximation property of neural networks, an adaptive neural network controller is presented by generalizing the model reference adaptive control technique, in which the gains of the controller are approximated by neural networks. This removes the requirement of linearizing the dynamics of the system, and the stability properties of the closed-loop system can be guaranteed.

1. Introduction

The LOS system is one that possesses the ability to maintain the line-of-sight of an electro-optical sensor when it is subjected to external disturbances. The presence of the flywheel, which is spinning at some specified high speed, is used to provide a high angular momentum, which maintains a fixed orientation in inertia space in the absence of external torques. This, however, causes the movements between the two axes of the LOS system to be highly coupled. In the past, research has been done to design controllers that decouple the axes of the LOS system. These included the use of model reference adaptive control [1], and a fuzzy decoupler [2]. The model reference adaptive control requires the model of the system to be linearized about the origin [1]. This

* Corresponding author

PII: S0957–4158(98)00030–0

may not be desirable if the required operating points are large. On the other hand, the fuzzy decoupling controller [2] requires much tuning to be done before desirable results may be obtained. In addition, there is no stability proofs for the fuzzy controller proposed in [2].

In this paper, a novel adaptive neural network feedback controller is proposed. We borrow the ideas from model reference adaptive control for linear systems [3, 4] and incorporate the use of neural networks in the adaptive control of a nonlinear system. By the use of Gaussian Radial Basis Function (GRBF) networks, the control of the nonlinear plant system may be transformed into a linear problem. Thus, we are able to use the well-known adaptation laws for linear systems to solve the problem. In the proposed method, we require only an estimate of the system. In addition, the asymptotic stability of the proposed neural network controller has been proven theoretically using the Lyapunov Theorem.

In this paper, we will first give a brief review of neural networks and its approximation property. We will then introduce the passive LOS stabilization system, its working principles and its dynamics. Next, we introduce the adaptive neural network full state feedback controller. Simulation studies of the proposed controller are carried out on the LOS system to demonstrate its effectiveness.

2. Neural network approximation

The neural network is usually used in control engineering as a function approximator that emulates any given nonlinear function $f(x)$, under mild assumptions, up to any desired accuracy. There are two main classes of neural networks [5], namely the Multi-Layer Perceptron (MLP) and the Associative Memory Network (AMN).

The MLP is widely used in classification tasks with large feature spaces. Usually, extensive training periods are required for the MLP to learn the function to be approximated [5]. This makes it unsuitable for use in adaptive systems, where fast convergence is required. Hence, in this paper, we will concentrate mainly on a class of AMN, namely the GRBF.

It has been shown in [6] that a linear combination of GRBFs results in an optimal mean square approximation of an unknown function which is infinitely differentiable and whose values are specified at a finite set of points in $\Re^n$. It has been proven further in [7, 8] that any continuous functions, not necessarily infinitely smooth ones, can be uniformly approximated by a linear combination of GRBFs. In addition, the GRBF network belongs to a class of linearly parametrized models, which makes it suitable for constructing adaptation mechanisms within them. Moreover, the fast initial learning of the GRBF [5] also adds to its suitability in adaptive systems. Thus, the GRBF network is chosen in implementing the proposed controller.

The schematic diagram of the GRBF neural network is shown in Fig. 1. The GRBF consists of l numbers of Gaussian functions (nodes) with input variables $x \in \Re^n$; and the centre vector for each node is $\mu_i = (\mu_{i1}, \ldots, \mu_{in}) \in \Re^n$, where $i = 1, 2, \ldots, l$. A GRBF neural network consists of three layers, namely, the input layer, the hidden layer, where the GRBFs are, and the output layer. At the input layer, the input space is

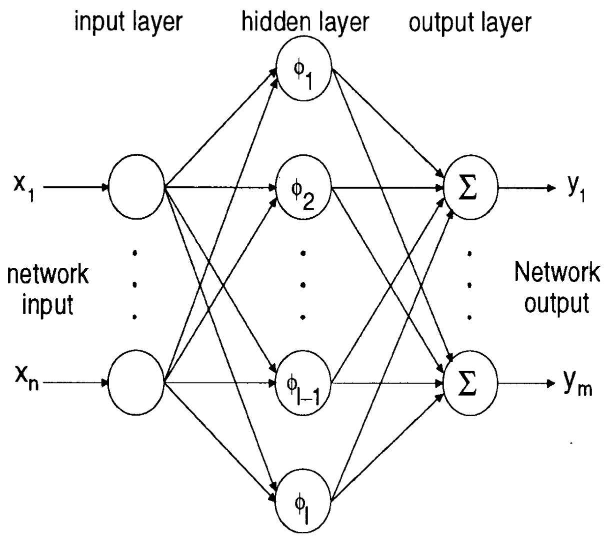

Fig. 1. GRBF neural network.

divided into grids with a localized Gaussian function at each node defining the receptive field in $\Re^n$, with centre μ_i and variance σ_i^2. How well the network models a given function is determined by the position of these centres and the variances of the GRBFs. The value of σ_i^2 defines the amount of overlap between adjacent GRBFs. The output layer is simply a linear combination of the outputs of the GRBFs. Then, for any given function $y = f(x)$, the results from [8] states that it can be approximated by the GRBF neural network. This can be expressed as

$$y = \hat{f}(W, x) = W^T \phi(x) \tag{2.1}$$

where $y \in \Re^m$, $W \in \Re^{k \times m}$ and $\phi(x) \in \Re^k$ and ϕ_i is given by

$$\phi_i(x) = \exp\left(-\frac{\|x-\mu_i\|_2^2}{\sigma_i^2}\right)$$

$$= \exp\left(-\frac{(x-\mu_i)^T(x-\mu_i)}{\sigma_i^2}\right) \tag{2.2}$$

It is thus clear from (2.1) that the output of the GRBF network is the linear combination of the outputs of the GRBFs (2.2), where W are the weights of the network to be determined.

In this paper, the centres of the GRBF, μ_i, are evenly spaced throughout the input space network. This ensures that the GRBF network covers the entire workspace of the system. The variance, σ_i^2, are chosen such that there is sufficient overlap between the basis functions.

3. Passive LOS stabilization system

The LOS system is one that possesses the ability to maintain the LOS of an electro-optical sensor when it is subjected to external disturbances such as the base motion of host vehicles on which it is mounted [9, 10]. Figure 2 shows a schematic diagram of the passive LOS stabilization system considered in this paper. The system is made up of the following components:

(1) a flywheel and its drive;
(2) gimbals that provide two degrees-of-freedom to the flywheel and torque motors for slewing purposes; and
(3) a mirror that is geared to the gimbal through a 2 : 1 reduction drive mechanism.

The flywheel, which is spinning at some specified high speed, is used to provide a high angular momentum. This angular momentum maintains a fixed orientation in inertial space when there is no external torques present, thus providing a directional

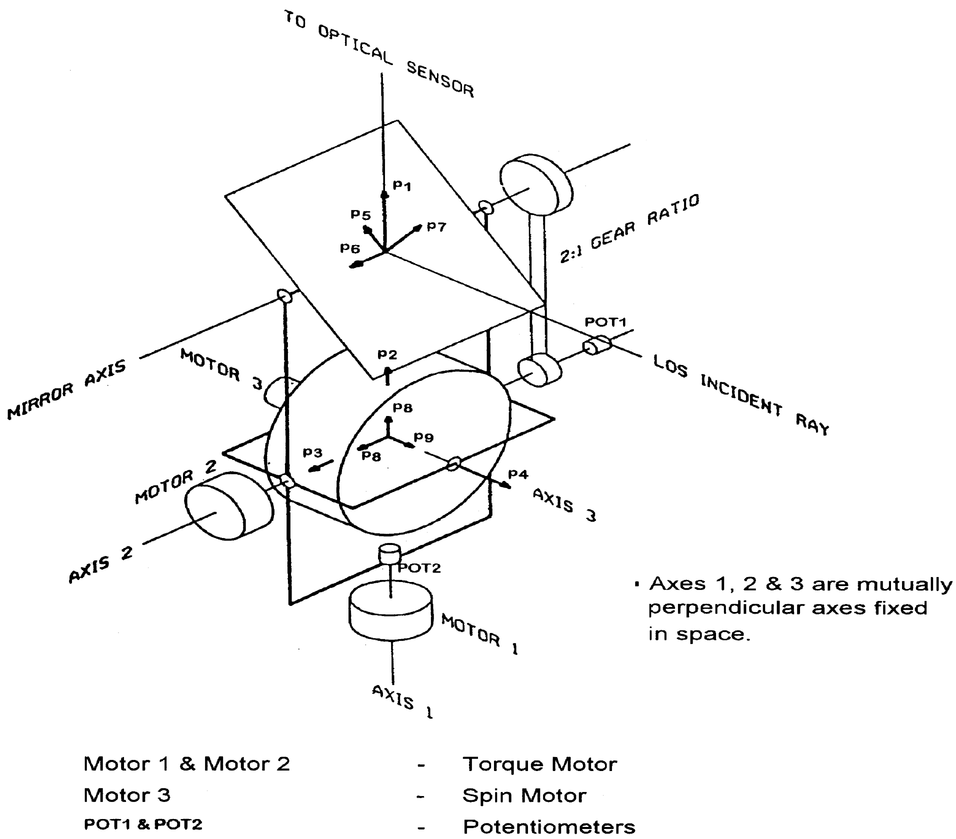

Fig. 2. Schematic of passive LOS stabilization system.

reference. The flywheel axis is mounted onto two frames called the inner and outer gimbals. This allows it to have two degrees-of-freedom. The flywheel is directly connected to the inner gimbal through two supporting bearings. The inner gimbal is then supported by the outer gimbal structure and is connected by a 2:1 reduction drive to the mirror mechanism. This 2:1 reduction drive is necessary because when the mirror is tilted by an angle α, the reflected LOS is rotated by 2α. The mirror's axis of rotation is parallel to the inner gimbal axis, and orthogonal to the outer gimbal axis.

Torquer motors are attached to the inner and outer gimbals to enable the system to accept steering commands and to correct for sightline drift. By providing an appropriate torque through the torquer motors, the system can be precessed relative to the inertial space to achieve some desired LOS. Once the input torque is removed, the LOS will now be space-stabilized (due to the property of the flywheel gyroscope) at its new position.

The dynamical model of the LOS stabilization system is given by the following equations [11],

$$d_1(\theta)\ddot{\theta}_1 + c_1(\theta, \dot{\theta}) = u_1 \tag{3.1}$$

$$d_2(\theta)\ddot{\theta}_2 + c_2(\theta, \dot{\theta}) = u_2 \tag{3.2}$$

where $\theta = [\theta_1, \theta_2]^T$ with θ_1 and θ_2 being the angular position of axes 1 and 2, respectively, as shown in Fig. 2; and $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2]^T$ are the corresponding angular velocities; u_1 and u_2 are the actuator torques for axes 1 and 2, respectively. The functions in eqns (3.1)–(3.2) are defined by

$$d_1(\theta) = p_1 + p_4 + (p_2 - p_4 + p_8)\cos^2\theta_2 + \tfrac{1}{2}(p_5 + p_7) + \tfrac{1}{2}(p_5 - p_7)\sin\theta_2 \tag{3.3}$$

$$\begin{aligned} c_1(\theta, \dot{\theta}) = & -(p_2 - p_4 + p_8)\dot{\theta}_1\dot{\theta}_2 \sin 2\theta_2 + \tfrac{1}{2}(p_5 - p_7)\dot{\theta}_1\dot{\theta}_2\cos\theta_2 \\ & + p_9\dot{\theta}_1\dot{\theta}_2 \sin\theta_2\cos\theta_2 + p_9\dot{\theta}_2\dot{\theta}_3\cos\theta_2 \end{aligned} \tag{3.4}$$

$$d_2(\theta) = p_3 + \frac{p_6}{4} + p_8 \tag{3.5}$$

$$\begin{aligned} c_2(\theta, \dot{\theta}) = & \tfrac{1}{2}(p_2 - p_4 + p_8)\dot{\theta}_1^2 \sin 2\theta_2 - \tfrac{1}{4}(p_5 - p_7)\dot{\theta}_1^2\cos\theta_2 \\ & - p_9\dot{\theta}_1^2 \sin\theta_2\cos\theta_2 - p_9\dot{\theta}_1\dot{\theta}_3\cos\theta_2 \end{aligned} \tag{3.6}$$

where p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8 and p_9 are all physical constants representing the various moments of inertia of the system; and $\dot{\theta}_3$ is the velocity of the flywheel, which is a large constant.

From the dynamic eqns (3.1)–(3.2), (3.3)–(3.6), the following properties of the LOS system may be inferred.

Property 1. The terms d_i, $i = 1, 2$ are positive definite. This is an essential property of the system. This property will be used later in the development of controllers.

Property 2. From (3.4)–(3.6), cross-couplings between the axes are introduced due to $\dot{\theta}_2$ terms appearing in (3.4) and $\dot{\theta}_1$ terms appearing in (3.6). The magnitudes of these values are small, and thus the cross-coupling effects are weak.

However, the inclusion of the flywheel introduces strong cross couplings between the axes of the system as can be seen from the presence of $\dot{\theta}_3$ in the last terms of eqns (3.4) and (3.6). $\dot{\theta}_3$ is usually in the order of thousand rpm. This strong cross-coupling increases the difficulty of the control problem.

For the passive LOS stabilization system, the control requirements can be stated as:

(1) to achieve a sufficiently high bandwidth with no steady-state error for step inputs and
(2) to decouple the system such that there is minimal cross-coupling effect in the system.

The ability of the proposed control to meet the above requirements will be considered in Section 5.

4. Direct neural networks model reference adaptive control

The dynamical eqns (3.1)–(3.2) can be transformed to a second-order nonlinear plant expressed in the controllability canonical form:

$$\dot{x}_1 = x_2 \tag{4.1}$$

$$\dot{x}_2 = f(x_1, x_2) + G(x_1, x_2)u \tag{4.2}$$

where $x_1 = [\theta_1 \quad \theta_2]^T \in \Re^2$, $x_2 = [\dot{\theta}_1 \quad \dot{\theta}_2]^T \in \Re^2$, $u = [u_1 \quad u_2]^T \in \Re^2$,

$$f(x_1, x_2) = \begin{bmatrix} -c_1(x_1, x_2)/d_1(x_1, x_2) \\ -c_2(x_1, x_2)/d_2(x_1, x_2) \end{bmatrix} \in \Re^2 \tag{4.3}$$

and

$$G(x_1, x_2) = \begin{bmatrix} g_1(x_1, x_2) & 0 \\ 0 & g_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} 1/d_1(x_1, x_2) & 0 \\ 0 & 1/d_2(x_1, x_2) \end{bmatrix} \in \Re^{2 \times 2} \tag{4.4}$$

For simplicity of presentation, we shall define $0_{m \times n} \in \Re^{m \times n}$ as an $m \times n$ zero matrix, $I_{n \times n} \in \Re^{n \times n}$ as an identity matrix of dimension $n \times n$; and let $x = [x_1^T \quad x_2^T]^T \in \Re^4$.

We can then cast the system equations into the following form

$$\dot{x} = A(t)x + \sum_{i=1}^{2} g_i(t) b_i u_i \tag{4.5}$$

where

$$A(t) = \begin{bmatrix} 0_{2\times2} & I_{2\times2} \\ H & M \end{bmatrix} \in \Re^{4\times4}$$

$$b_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad b_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \in \Re^4$$

with

$$H = 0_{2\times2} \tag{4.6}$$

$$M = \begin{bmatrix} 0 & m_{12} \\ m_{21} & 0 \end{bmatrix} \tag{4.7}$$

where

$$m_{12} = -\frac{1}{d_1}\left[-(p_2-p_4+p_8)\dot{\theta}_1 \sin 2\theta_2 + \frac{1}{2}(p_7-p_5)\dot{\theta}_1 \cos\theta_2 + p_9\dot{\theta}_1 \sin\theta_2 \cos\theta_2 + p_9\dot{\theta}_3 \cos\theta_2\right] \tag{4.8}$$

$$m_{21} = -\frac{1}{d_2}\left[\frac{1}{2}(p_2-p_4+p_8)\dot{\theta}_1 \sin 2\theta_2 - \frac{1}{4}(p_7-p_5)\dot{\theta}_1 \cos\theta_2 - p_9\dot{\theta}_1 \sin\theta_2 \cos\theta_2 - p_9\dot{\theta}_3 \cos\theta_2\right] \tag{4.9}$$

Note that for the case of the LOS system, the matrix H is a zero matrix. However, for other systems, matrix H may not necessarily be zero. An example of such a system is a prismatic joint robot. In this paper, we shall formulate the controller for the general case that H is non-zero.

Because $g_i(x) \neq 0$, we can always find control laws

$$u_i = k^{*T}_i x + m^*_i r_i \tag{4.10}$$

where $k^*_i \in \Re^4$ and $m^*_i \in \Re$, $i = 1, 2$, such that the closed-loop system matches the stable reference model:

$$\dot{x}_m = A_m x_m + \sum_{i=1}^{2} b_{mi} r_i \tag{4.11}$$

where $x_m \in \Re^4$ and

$$A_m = \begin{bmatrix} 0_{2\times2} & I_{2\times2} \\ H_m & D_m \end{bmatrix} \in \Re^{4\times4}$$

$$b_{m1} = \begin{bmatrix} 0 \\ 0 \\ v_{m1} \\ 0 \end{bmatrix}, \quad b_{m2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ v_{m2} \end{bmatrix} \in \Re^4$$

where $H_m = [h_{mij}] \in \Re^{2\times 2}$ and $D_m = [d_{mij}] \in \Re^{2\times 2}$. This can be seen by substituting the control laws (4.10) into the system eqn (4.5). The closed-loop system is given by

$$\dot{x} = Ax + \sum_{i=1}^{2} g_i b_i (k^{*T}_i x + m^*_i r_i)$$

$$= \left(A + \sum_{i=1}^{2} g_i b_i k^{*T}_i\right) x + \sum_{i=1}^{2} g_i b_i m^*_i r_i \tag{4.12}$$

Comparing the closed-loop system (4.12) to match the reference model (4.11), we get

$$A + \sum_{i=1}^{2} g_i b_i k^{*T}_i = A_m \tag{4.13}$$

$$g_i b_i m^*_i = b_{mi} \tag{4.14}$$

By equating the last two rows of the matrix eqn (4.13), we get

$$k^{*T}_i = g_i^{-1}([h_{mi} \quad d_{mi}] - [h_i \quad d_i]) \tag{4.15}$$

where h_i, d_i, h_{mi} and d_{mi} denotes the i-th rows of the respective matrices H, D, H_m and D_m. Similarly, we can derive an expression for m^*_i by equating the last two rows of the vector eqn (4.14).

$$m^*_i = g_i^{-1} v_{mi} \tag{4.16}$$

The difficulty of the control problem lies in the fact that we do not know exactly the matrices H and D. Thus, adaptive mechanisms can be used to solve the problem. However, from (4.15)–(4.16), it is obvious that k^*_i and m^*_i must all be functions of t, and the usual adaptation laws for linear systems [3, 4] cannot be applied in this case. Under the assumptions that H and D are unknown, k^*_i and m^*_i are unknown either. From (4.15)–(4.16), we can only guarantee their existence and smoothness. By noting that the neural network is a universal approximator, we can then express the controller gains by their respective neural networks approximations, i.e.

$$k^*_i = [\{W^*_{ki}\}^T \cdot \{\Xi_{ki}\}] + \varepsilon_{ki}, \tag{4.17}$$

$$m^*_i = [\{W^*_{mi}\}^T \cdot \{\Xi_{mi}\}] + \varepsilon_{mi}, \tag{4.18}$$

where $\{*\}$ are GL matrices and "$\cdot$" is the GL operator [12–14], $\{W^*_{ki}\}$ and $\{W^*_{mi}\}$ are the ideal weights of the neural networks, $\{\Xi_{ki}\}$, $\{\Xi_{mi}\}$ are the outputs of the bounded basis functions, and $\varepsilon_{ki} \in \Re^4$ and $\varepsilon_{mi} \in \Re$ are the neural networks approximation errors. For a fixed number of nodes, we know that $\|\varepsilon_{ki}\|$ and $|\varepsilon_{mi}|$ are bounded, W^*_{ki} and W^*_{mi} are unknown bounded constant parameters.

The GL operator is a tool that allows us to express the neural networks mathematically, to facilitate in the analysis of the neural networks. The interested reader may refer to [12–14] for more information on GL matrices and operators.

We consider the following control law with time-varying neural network weights

$$u_i = [\{\hat{W}_{ki}(t)\}^T \cdot \{\Xi_{ki}(x)\}]x + \{\hat{W}_{mi}(t)\}^T \cdot \{\Xi_{mi}(x)\} r_i + u_{rbi} \tag{4.19}$$

where u_{rbi}, $i = 1, 2$ are defined later to compensate for the approximation errors of the neural networks; $\{\hat{W}_{ki}(t)\}$ and $\{\hat{W}_{mi}(t)\}^T$ are the estimates of $\{W^*_{ki}\}$ and $W^*_{mi}\}$.

Define

$$e = x - x_m \tag{4.20}$$

$$\{\tilde{W}_{ki}\} = \{\hat{W}_{ki}\} - \{W^*_{ki}\}, \tag{4.21}$$

$$\{\tilde{W}_{mi}\} = \{\hat{W}_{mi}\} - \{W^*_{mi}\}, \tag{4.22}$$

where $i = 1, 2$.

Substituting the control laws (4.19) into the system eqn (4.5), and using (4.21)–(4.22), we get

$$\begin{aligned}
\dot{e} &= \dot{x} - \dot{x}_m \\
&= \left(A + \sum_{i=1}^{2} g_i b_i [\{\hat{W}_{ki}\}^T \cdot \{\Xi_{ki}\}]\right) x + \sum_{i=1}^{2} g_i b_i \{\hat{W}_{mi}\}^T \cdot \{\Xi_{mi}\} r_i \\
&\quad - A_m x_m - \sum_{i=1}^{2} b_{mi} r_i + \sum_{i=1}^{2} g_i b_i u_{rbi} \\
&= \left(A + \sum_{i=1}^{2} g_i b_i [\{W^*_{ki}\}^T \cdot \{\Xi_{ki}\}]\right) + \sum_{i=1}^{2} g_i b_i \{W^*_{mi}\}^T \cdot \{\Xi_{mi}\} r_i - A_m x_m - \sum_{i=1}^{2} b_{mi} r_i \\
&\quad + \sum_{i=1}^{2} g_i b_i [\{\tilde{W}_{ki}\}^T \cdot \{\Xi_{ki}\}] x + \sum_{i=1}^{2} g_i b_i \{\tilde{W}_{mi}\}^T \cdot \{\Xi_{mi}\} r_i + \sum_{i=1}^{2} g_i b_i u_{rbi}
\end{aligned}$$

By noting the neural network approximations (4.17)–(4.18), we obtain the following error equation

$$\begin{aligned}
\dot{e} &= \left(A + \sum_{i=1}^{2} g_i b_i k^*\right) x + \sum_{i=1}^{2} g_i b_i m^*_i r_i - A_m x_m - \sum_{i=1}^{2} b_{mi} r_i \\
&\quad + \sum_{i=1}^{2} g_i b_i [\{\tilde{W}_{ki}\}^T \cdot \{\Xi_{ki}\}] x + \sum_{i=1}^{2} g_i b_i [\{\tilde{W}_{mi}\}^T \cdot \{\Xi_{mi}\}] r_i \\
&\quad + \sum_{i=1}^{2} g_i b_i u_{rbi} - \sum_{i=1}^{2} g_i b_i (\varepsilon_{ki} x + \varepsilon_{mi} r_i)
\end{aligned} \tag{4.23}$$

Substituting A_m and b_{mi} from (4.13)–(4.14) into (4.23) leads to

$$\dot{e} = A_m e + \sum_{i=1}^{2} g_i b_i [\{\tilde{W}_{ki}\}^T \cdot \{\Xi_{ki}\}] x + \sum_{i=1}^{2} g_i b_i \{\tilde{W}_{mi}\} \cdot \{\Xi_{mi}\} r_i + \sum_{i=1}^{2} g_i b_i u_{rbi} - \sum_{i=1}^{2} g_i b_i (\varepsilon_{ki} x + \varepsilon_{mi} r_i) \quad (4.24)$$

Let P be the symmetric positive definite solution of the Lyapunov equation

$$PA_m + A_m^T P = -Q \quad (4.25)$$

where Q is symmetric positive definite.

In this paper, we shall assume that $\bar{g}_i$, the estimate of g_i is known for controller construction. By defining

$$\tilde{g}_i = g_i - \bar{g}_i \quad (4.26)$$

we propose the following adaptive laws to update the weights of the neural network on-line for closed-loop stability.

Theorem 4.1. For the nonlinear system (4.1)–(4.2), consider the neural network based control laws (4.19). If the adaptive laws for updating the weights of the adaptive neural network are given by

$$\{\dot{\hat{W}}_{ki}\} = -\bar{g}_i e^T P b_i \{\Gamma_{ki}\} \cdot \{\Xi_{ki}\} x \quad (4.27)$$

$$\{\dot{\hat{W}}_{mi}\} = -\bar{g}_i e^T P b_i \{\Gamma_{mi}\} \cdot \{\Xi_{mi}\} r_i \quad (4.28)$$

and

$$u_{rbi} = -K_i \operatorname{sgn}(e^T P b_i) \quad (4.29)$$

$$K_i \geqslant |g_i(\varepsilon_{ki} x + \varepsilon_{mi} r_i) + \tilde{g}_i \{\tilde{W}_{ki}^T\} \cdot \{\Xi_{ki}\} x + \tilde{g}_i \{\tilde{W}_{mi}^T\} \cdot \{\Xi_{mi}\} r_i| \quad (4.30)$$

where $i = 1, 2$ and Γ_{ki}, Γ_{mi} are dimensionally compatible positive definite matrix, then the adaptive neural network controller ensures that all states of the system will be bounded, and in addition,

$$\lim_{t \to \infty} (x - x_m) = 0 \quad (4.31)$$

i.e., asymptotic tracking is achieved.

Proof: The proof of the theorem can be completed by following the same procedure as discussed in [15] for a class of general nonlinear systems. It is given here for completeness.

Choose the following Lyapunov function candidate

$$V = e^T Pe + \sum_{i=1}^{2} \tilde{W}_{ki}^T \Gamma_{ki}^{-1} \tilde{W}_{ki} + \sum_{i=1}^{2} \tilde{W}_{mi}^T \Gamma_{mi}^{-1} \tilde{W}_{mi} \tag{4.32}$$

where Γ_{ki}, Γ_{mi} are positive symmetric definite matrices of the appropriate dimensions, and P is the positive symmetric definite matrix solution of the Lyapunov eqn (4.25).

Differentiating the Lyapunov function with respect to time, and substituting the error eqn (4.24) yields the following expression

$$\dot{V} = -e^T Qe + \sum_{i=1}^{2} 2g_i e^T Pb_i \{\tilde{W}_{ki}\} \cdot \{\Xi_{ki}\} x + \sum_{i=1}^{2} 2g_i e^T Pb_i \{\tilde{W}_{mi}\} \cdot \{\Xi_{mi}\} r_i$$

$$+ \sum_{i=1}^{2} 2\tilde{W}_{ki}^T \Gamma_{ki}^{-1} \dot{\hat{W}}_{ki} + \sum_{i=1}^{2} 2\tilde{W}_{mi}^T \Gamma_{mi}^{-1} \dot{\hat{W}}_{mi} + \sum_{i=1}^{2} 2g_i e^T Pb_i u_{rbi} - \sum_{i=1}^{2} 2g_i e^T Pb_i (\varepsilon_{ki} x + \varepsilon_{mi} r)$$

Substituting the adaptation laws (4.27)–(4.28) and u_{rbi} from (4.29)–(4.30), we have

$$\dot{V} \leqslant -e^T Qe \leqslant 0 \tag{4.34}$$

which results in uniform boundedness of e, and the boundedness of $\tilde{W}_i$, i.e. the boundedness of $\hat{W}_i$. Furthermore, from (4.24), $\dot{e}$ is bounded. Hence, we can conclude that

$$\lim_{t \to \infty} \|e\| = 0 \tag{4.35}$$

which in turn implies that

$$\lim_{t \to \infty} (x(t) - x_m(t)) = 0 \tag{4.36}$$

i.e., asymptotic tracking is achieved.

From (4.29)–(4.30), we find that the sliding mode gains, K_i of the control law depends not only on the neural network approximation errors, ε_{ki} and ε_{mi}, but also on the states of the system, x, and the reference commands, r_i. This leads to time varying sliding mode gains. On the other hand, if we require a constant sliding mode gain, then K_i has to be chosen sufficiently large. However, we know that a large sliding mode gain introduces more chattering in the control signal. Thus, we present the following corollary, which results in the same stability properties, where the gains of the sliding mode control terms are independent of the state x and the reference commands r_i.

Corollary 1. For the nonlinear system (4.1)–(4.2), consider the neural network based control laws (4.19). If the adaptive laws for updating the weights of the adaptive network are given by (4.27)–(4.28), and

$$u_{rbi} = -K_{1i} \operatorname{sgn}(e^T Pb_i) x - K_{2i} \operatorname{sgn}(e^T Pb_i) r_i - K_{3i} \operatorname{sgn}(e^T Pb_i) x - K_{4i} \operatorname{sgn}(e^T Pb_i) r_i \tag{4.37}$$

$$K_{1ij} \geqslant |\varepsilon_{kij}| \tag{4.38}$$

$$K_{2i} \geqslant |\varepsilon_{mi}| \tag{4.39}$$

$$K_{3ij} \geqslant |\tilde{g}_i\{\tilde{W}^T_{kij}\}\cdot\{\Xi_{kij}\}| \tag{4.40}$$

$$K_{4i} \geqslant |\tilde{g}_i\{\tilde{W}^T_{mi}\}\cdot\{\Xi_{mi}\}| \tag{4.41}$$

where K_{1ij} and K_{3ij} denote the individual elements of the vectors K_{1i} and K_{3i}, respectively. Using the adaptation laws (4.27)–(4.28), the adaptive neural network controller ensures that all states of the system will be bounded, and in addition,

$$\lim_{t\to 0}(x - x_m) = 0 \tag{4.42}$$

i.e., asymptotic tracking is achieved.

Proof: Choosing the same Lyapunov function candidate (4.32), and differentiating with respect to time, we obtain the following expression

$$\dot{V} = -e^T Qe + \sum_{i=1}^{2} 2g_i e^T Pb_i\{\tilde{W}_{ki}\}\cdot\{\Xi_{ki}\}x + \sum_{i=1}^{2} 2g_i e^T Pb_i\{\tilde{W}_{mi}\}\cdot\{\Xi_{mi}\}r_i$$
$$+ \sum_{i=1}^{2} 2\tilde{W}^T_{ki}\Gamma^{-1}_{ki}\dot{\hat{W}}_{ki} + \sum_{i=1}^{2} 2\tilde{W}^T_{mi}\Gamma^{-1}_{mi}\dot{\hat{W}}_{mi} + \sum_{i=1}^{2} 2g_i e^T Pb_i u_{rbi} - \sum_{i=1}^{2} 2g_i e^T Pb_i(\varepsilon_{ki}x + \varepsilon_{mi}r) \tag{4.43}$$

Substituting the adaptation laws (4.27)–(4.28) and u_{rbi} from (4.37)–(4.41), we have

$$\dot{V} \leqslant -e^T Qe \leqslant 0 \tag{4.44}$$

which results in uniform boundedness of e, and the boundedness of $\tilde{W}_i$, i.e. the boundedness of $\hat{W}_i$. From (4.24) $\dot{e}$ is bounded. Hence, we can conclude that

$$\lim_{t\to\infty}\|e\| = 0 \tag{4.45}$$

which in turn implies that

$$\lim_{t\to\infty}(x(t) - x_m(t)) = 0 \tag{4.46}$$

i.e., asymptotic tracking is achieved.

As can be seen from Corollary 1, the gains K_{1ij}, K_{2i}, K_{3ij} and K_{4i} in eqns (4.37)–(4.41) are lower bounded by the respective neural network approximation and adaptation errors, which are small. This is in contrast to the gains K_i of (4.29)–(4.30), which are time varying since they are dependent on the states and the input commands.

Hence, the sliding mode gains may be kept small due to the small neural network approximation errors.

5. Simulation studies

After formulating the neural network adaptation control laws, simulation studies were done to show the effectiveness of the scheme.

The reference model used in this simulation study is

$$\begin{bmatrix} \dot{\theta}_m \\ \ddot{\theta}_m \end{bmatrix} = \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} \\ H_m & D_m \end{bmatrix} \begin{bmatrix} \theta_m \\ \dot{\theta}_m \end{bmatrix} + \begin{bmatrix} 0_{2\times 1} \\ V_{m1} \end{bmatrix} r_1 + \begin{bmatrix} 0_{2\times 1} \\ V_{m2} \end{bmatrix} r_2 \tag{5.1}$$

where

$$H_m = \mathrm{diag}[-36], \quad D_m = \mathrm{diag}[-12] \tag{5.2}$$

$$V_{m1} = \begin{bmatrix} 36 \\ 0 \end{bmatrix}, \quad V_{m2} = \begin{bmatrix} 0 \\ 36 \end{bmatrix} \tag{5.3}$$

Note that the matrices H_m and D_m are chosen to be diagonal such that the outputs of both axes are decoupled. This meets the second control requirement stated in Section 3. Choosing

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \tag{5.4}$$

yield

$$P = \begin{bmatrix} 20.8 & 0.0 & -50.0 & 0 \\ 0.0 & 20.8 & 0.0 & -50.0 \\ -50.0 & 0.0 & 150.4 & 0.0 \\ 0.0 & -50.0 & 0.0 & 150.4 \end{bmatrix} \tag{5.5}$$

The nodes of the GRBF network are spaced evenly throughout the input space, with a total 625 nodes. This ensures that the entire operation region of the LOS system is being covered.

In the simulation studies, step changes in the command signals are used. The adaptation gain matrices, Γ_i are chosen to be $\Gamma_{k1} = \mathrm{diag}[0.1]$, $\Gamma_{k2} = \mathrm{diag}[0.1]$,

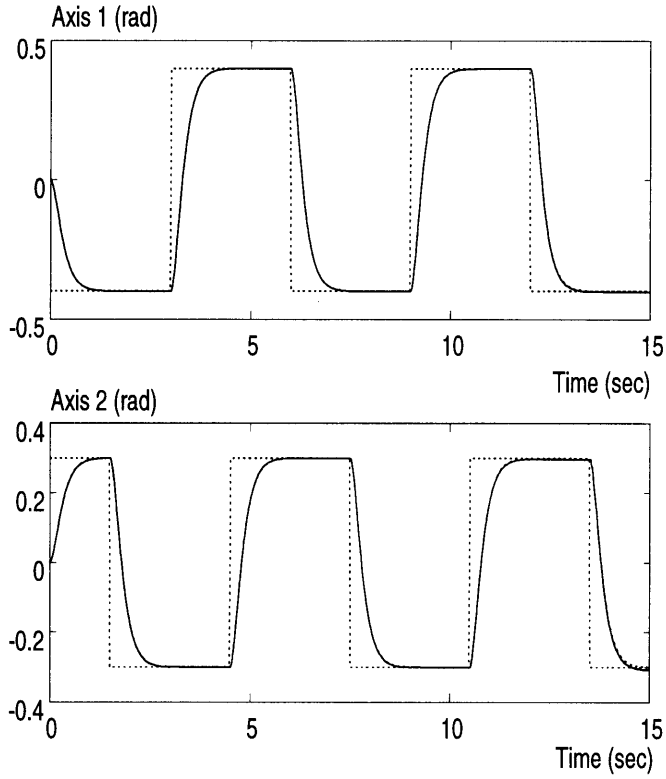

Fig. 3. Outputs of the passive LOS stabilization system.

$\Gamma_{m1} = \mathrm{diag}[40.0]$ and $\Gamma_{m2} = \mathrm{diag}[30.0]$. Figure 3 shows the output tracking performance of the system to the changes in the reference signals. The results show that the first requirement is met as the controller has been shown to achieve assured tracking.

In addition, results also show that the controller has successfully decoupled the cross-coupling effects between the axes, in the sense that changes in the reference signal of one axis does not affect the output of the other. The control effort and the controller gains (output of the neural networks) are given in Figs 4 and 5, respectively.

The numbers of nodes used in the GRBF network were chosen by trial and error through computer simulations. It was observed that with a smaller number of neural network nodes, say 81 nodes, the control performance is not that good. This could be due to the poor approximation of the neural network. When the number of neural network nodes is increased to 625, there is a significant improvement in the control

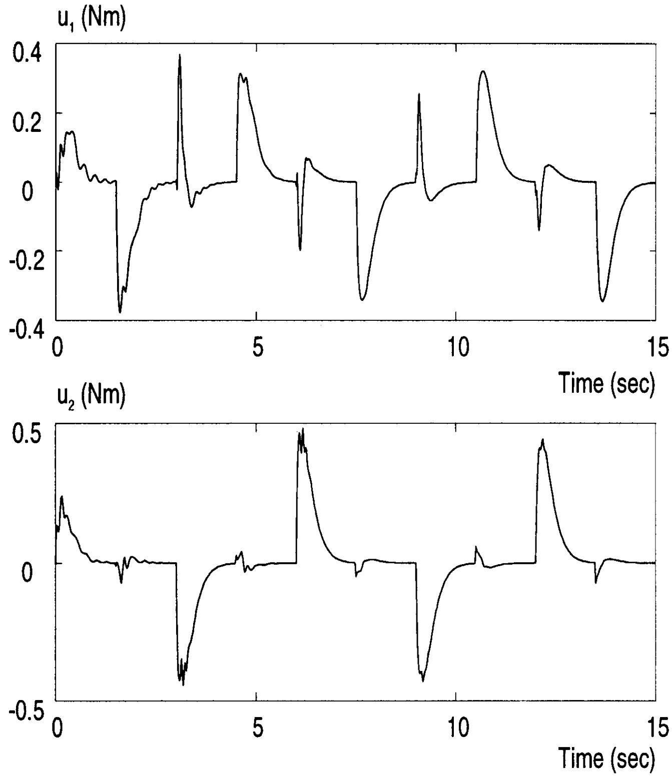

Fig. 4. Control signals.

performance. However, when it is increased further, there is no perceptible improvement in the control performance.

6. Conclusion

An adaptive neural network feedback control of a passive LOS stabilization system has been proposed in this paper. Although the system equations are nonlinear, we are able to exploit the universal approximation property of neural networks to transform the problem into a linear one. It has been shown that the proposed controller is able to achieve asymptotic tracking of the reference command signals. Simulation studies of the proposed neural network controller applied to the LOS stabilization system are also presented to demonstrate its effectiveness. Further work is direct at generalizing the results to include a wider class of nonlinear systems.

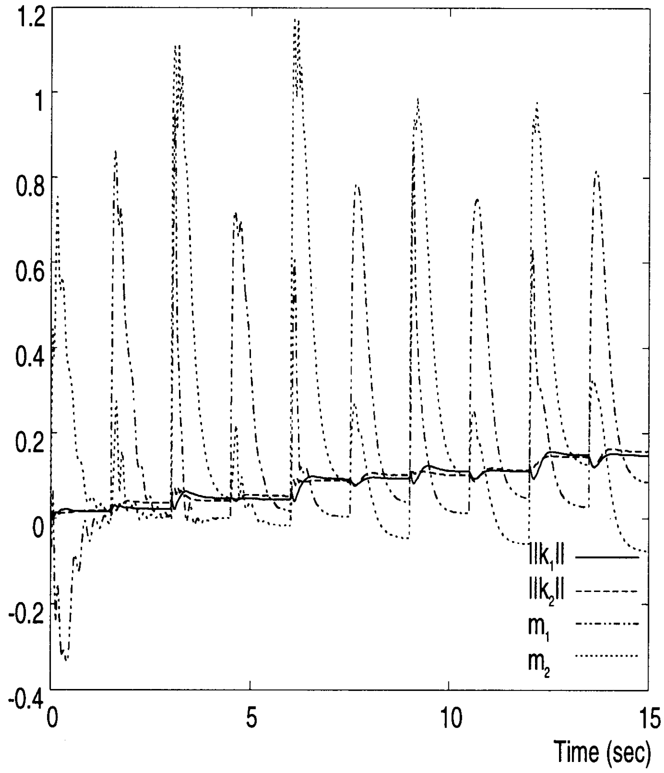

Fig. 5. Controller gains (outputs of the neural networks).

References

[1] Lee TH, Koh EK, Loh MK. Stable adaptive control of multivariable servomechanisms, with application to a passive line-of-sight stabilization system. IEEE Trans on Industrial Electronics 1996;43(1):98–105.

[2] Lee MW. An investigation in fuzzy logic. B. Eng. thesis, Department of Electrical Engineering, National University of Singapore, 1995.

[3] Åstöm KJ, Wittenmark B. Adaptive Control. Reading, MA: Addison-Wesley, 1995.

[4] Narendra KS, Annaswamy AM. Stable Adaptive Systems. Reading, MA: Addison-Wesley, 1989.

[5] Brown M, Harris C. Neurofuzzy Adaptive Modelling and Control. London: Prentice-Hall, 1994.

[6] Poggio T, Girosi F. A theory of networks for approximation and learning, Artificial Intelligence Lab. Memo, No. 1140. Cambridge, MA: MIT, July, 1989.

[7] Girosi F, Poggio T. Networks and the best approximation property, Artificial Intelligence Lab. Memo, No. 1164. Cambridge, MA: MIT, October, 1989.

[8] Poggio T, Girosi F. Networks for approximation and learning. Proc of IEEE 1990;78:1481–97.

[9] Alford DW. The development of a directional gyroscope for remotely piloted vehicles and similar applications. Mech. Tech. Inertial Devices, Proc. Inst. Mech. Eng. Newcastle, Australia, 1987;1–8.

[10] Bigley WJ, Rizzo VJ. Wideband linear quadratic control of a gyro-stabilized electro-optical sight system. IEEE Control System Magazine 1987;7(4):20–4.

[11] Loh MK. Design, development and control of a LOS stabilisation system. Master thesis, Department of Electrical Engineering, National University of Singapore, 1991.

[12] Ge SS, Postlethwaite I. Adaptive neural network controller design for flexible joint robots using singular perturbation technique. Trans Inst of Measurement and Control 1995;17(3):120–31.

[13] Ge SS, Lee TH. Robust model reference adaptive control of robots based on neural network parametrization. Proc of ACC, Albuquerque, NM, 1997;3:2006–10.

[14] Ge SS. Robust adaptive NN feedback linearization control of nonlinear systems. International Journal of Systems Science 1997;27(12):1327–38.

[15] Ge SS, Lee TH, Harris CJ. Adaptive Neural Network Control of Robotic Manipulators. World Scientific Publishing, Singapore, 1998, to be published.