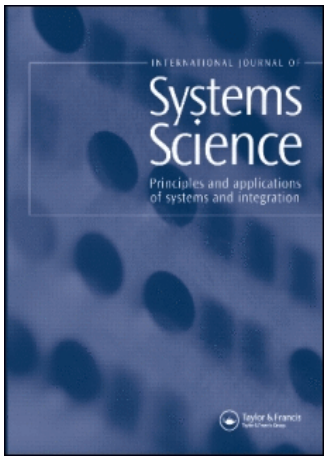


International Journal of Systems Science
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713697751

Real-time neural-network-based control of a gyro-mirror line-of-sight stabilization platform

T. H. Lee [a]; S. S. Ge [a]; C. C. Hang [a]; Q. Zhao [a]; A. S. Au [a]
[a] Department of Electrical Engineering, National University of Singapore. Singapore

To cite this Article: Lee, T. H., Ge, S. S., Hang, C. C., Zhao, Q. and Au, A. S. , 'Real-time neural-network-based control of a gyro-mirror line-of-sight stabilization platform', International Journal of Systems Science, 29:7, 745 - 758
To link to this article: DOI: 10.1080/00207729808929568
URL: http://dx.doi.org/10.1080/00207729808929568

International Journal of Systems Science, 1998, volume 29, number 7, pages 745–758

Real-time neural-network-based control of a gyro-mirror line-of-sight stabilization platform

T. H. LEE†, S. S. GE†‡, C. C. HANG†, Q. ZHAO† and A. S. AU†

In this paper, we consider the real-time neural-network-based control of a gyro-mirror line-of-sight stabilization platform. This is an example of a multivariable nonlinear servomechanism encountered in certain practical applications, and the appropriate neural-network-based modelling and controller design are discussed. In the paper, real-time experimental results in applying the proposed controller to a pilot-scale gyro-mirror platform are presented to demonstrate its effectiveness. These experiments also serve to verify the analytical results in a proptotype real-time application.

1. Introduction

The gyro-mirror line-of-sight (LOS) stabilization platform (see Fig. 1) is a very useful device which forms the basis of a wide range of practical instruments used for purposes of sighting and targeting in both surface and air-borne vehicles (Alford 1987, Bigley and Rizzo 1987). By virtue of its inertial properties, this line-of-sight stabilization platform possesses the ability to maintain the fixed direction of the sight line of an electro-optical sensor when it is subjected to external disturbance such as the base motion of host vehicles on which it is mounted. The dynamics of such a platform are highly nonlinear, and the difficulty of the problem is compounded by the need to operate at different gyroscope flywheel speeds and under different payload conditions. For control of such systems, traditional techniques which are commonly used include optimal control (Zhang and Barton 1991), conventional variable-structure control (Hashimoto 1987) and linear adaptive control (Sepe and Lang 1991). However, most of these lose effectiveness when the servomechanisms have fairly severe nonlinear dynamics and, more recently, there has been increasing research and development work to address this problem with the design of control systems, incorporating neural networks. Some examples have been given by Nguyen and Widrow (1990), Ozaki *et al.* (1991), Tzirkel and Fallside (1991), Lee *et al.* (1992), Sanner and Slotine (1992), Lee and Tan (1993) and Ge *et al.* (1997), but there are undoubtedly other interesting and related developments. In these, neural networks are used in various ways to construct nonlinear control systems to handle a variety of control objectives. The key elements in the design procedure are the proper choice of the functions to be approximated and an appropriate control strategy that utilizes this approximation.

In this paper, we further extend the earlier work (Lee *et al.* 1992, Lee and Tan 1993, Ge *et al.* 1997), that we have carried out in neural-network-based servomechanism control systems and develop a real-time neural-based controller applicable to the gyro-mirror LOS stabilization platform. Note that this is a multivariable servomechanism with nonlinear dynamics and, in the developments presented in this paper, adaptive techniques are utilized and two types of radial basis function (RBF) neural networks are considered: the Gaussian RBF and the Hardy multiquadric RBF. The appropriate neural-network-based modelling and controller design are discussed in the paper and, further, real-time experimental results in applying the proposed controller to a pilot-scale gyro-mirror platform are also presented to demonstrate its effectiveness. These experiments also serve to verify the analytical results in a prototype real-time application.

2. Radial basis function neural networks

The function approximation problem is concerned with the problem of approximating or interpolating a continuous multivariate function $f(x) : R^n \rightarrow R^m$ where $x = (x_1, x_2, \ldots, x_n)$ by an approximating function

Received 30 December 1996. Accepted 30 October 1997.

† Department of Electrical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260.

‡ Author to whom all correspondence should be addressed.

$F(W,x) : R^s \times R^n \rightarrow R^m$ having a fixed number of parameters W with $W = [w_1, w_2, \ldots, w_s]^T$. Some metric (or distance function) p may be used to measure the distance of $p[f(x), F(W,x)]$ of an approximation $F(W,x)$ from $f(x)$. The distance is usually in the form of a norm, for instance the L_2 norm. The approximation problem can then be stated formally (Rice 1964) as follows.

Definition: If $f(x)$ is a continuous function defined on set X and $F(W,x)$ is an approximating function that depends continuously on W and x, the *approximation problem* is to determine the optimal parameters W^* such that

$$p[F(W^*,x), f(x)] \leq \epsilon$$

for an acceptably small ϵ. □

A solution to this problem, if it exists and corresponds to the smallest ϵ, is said to be the best approximation. The existence of a best approximation depends ultimately on the class of functions to whom $F(W,x)$ belongs. The learning algorithm is then to minimize the distance p. Hence, there are two distinct problems in functions approximation. The first problem is the *representation problem* which deals with which approximating function $F(W,x)$ can best approximate the function $f(x)$. The other problem is the *learning problem*, which is to find the training method to ensure that the optimal parameters W^* for a given choice of F is obtained.

Because of its 'learning' ability, a neural network (Khanna 1990) is well suited to provide the approximate function $F(W,x)$ and, among the different classes of neural network structures that may be considered, there is a particular class that utilizes RBFs in its internal structure. For this class, the RBFs are those functions in the hidden units (Khanna 1990) which constitute an arbitrary 'basis' for the input vectors when they are expanded into the hidden layer. A typical RBF neural network thus has three different layers, namely the input layer, the hidden layer and the output layer, and, as indicated previously, it is the hidden layer that contains the RBFs. The transformation from input space to hidden-unit space is *nonlinear*, and the transformation from the hidden-unit space to the output space is *linear*. The use of these RBFs in the design of artificial neural networks has been discussed by Broomhead and Lowe (1988) for example, and the approximation of continuous functions by RBF neural networks has been analysed by Poggio and Girosi (1990). A number of function types can be used for the RBFs, and these would include the Gaussian function class and the Hardy multiquadric functions class (Micchelli 1986). In the work presented in this paper, we shall consider these two kinds of RBF because the Gaussian RBFs and the Hardy multiquadric RBFs both have some attractive properties when used as basis functions in the approximation problem (Micchelli 1986). While the Hardy multiquadric does not depend strongly on the distribution of data points (Franke 1982), Gaussian RBFs have other attractive properties, such as firstly they are bounded, strictly positive and absolutely integrable on R^n, secondly their Fourier transforms are their own and thirdly they are time–frequency localized (Sanner and Slotine 1992).

Thus for the function $y = f(x)$, $x \in R^n$, the appropriate approximation expressed in terms of the RBF neural networks is

$$\hat{y} = F(W,x) = W^T\phi(x), \quad \text{where } W,\ \phi(x) \in R^m, \tag{2.1}$$

where W and $\phi(x)$ are the weight and RBF vectors respectively, and m is the number of nodes. If Gaussian RBFs are used, then the elements of the RBF vector are given by

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

and, if the Hardy multiquadric RBFs are used, then

$$\phi_i(x) = [\sigma_i^2 + (x-c_i)^T(x-c_i)]^{1/2}. \tag{2.3}$$

At the input layer, the multidimensional input space is divided into grids with a radial basis function at the ith node defining a receptive field in R^n with $c_i \in R^n$ as its centre's position vector and $\sigma_i^2 \in R$ as its variance. The centre vector c_i defines the centre's position in the input space which can be set arbitrarily. The value of σ_i^2 is essentially the width of the ith node which can be different from other nodes. The approximation capability of the Gaussian and the Hardy multiquadric RBF neural networks will be utilized in the work described in this paper for the modelling and controller design of the gyro-mirror platform.

3. Gyro-mirror line-of-sight stabilization platform

Fig. 1 shows schematically the LOS stabilization platform considered in this paper. The system consists of the following components:

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

The flywheel spinning at some specified highspeed is used to provide a high angular momentum. As the angular momentum is a vector, it will maintain a fixed orientation in inertial space when it is not subjected to

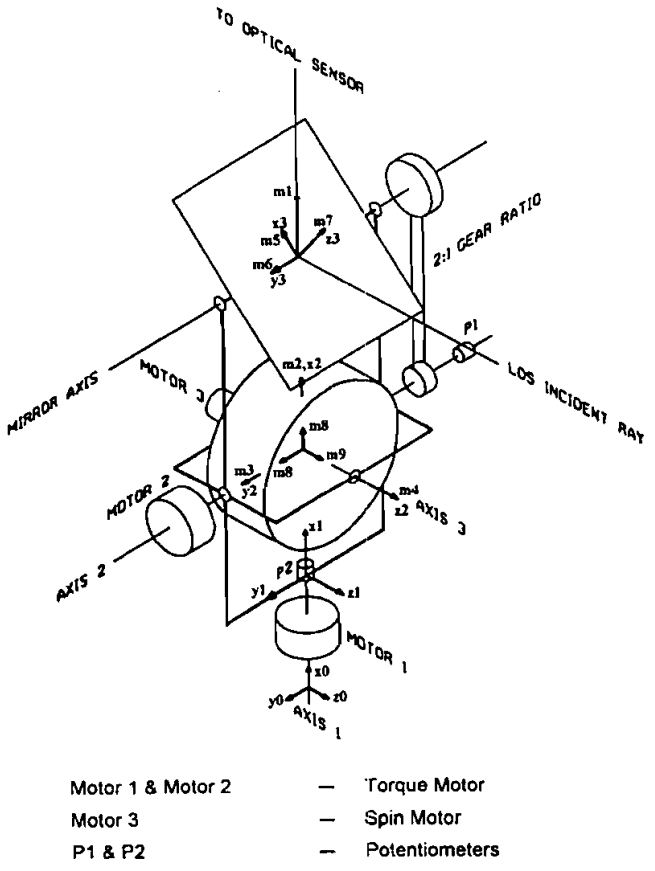


Figure 1. Schematic diagram of passive LOS stabilization system.

external torques, thereby providing a directional reference. For the flywheel axis to have two degrees of freedom, it is mounted onto two frames called the inner and outer gimbals. 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. The mirror's axis of rotation is parallel to the inner gimbal axis, and orthogonal to the outer gimbal axis. The 2 : 1 reduction drive is required because, when the mirror is tilted by an angle α, the reflected sight line is rotated by 2α.

A torque motor is attached to each of the inner and the outer gimbals. These torque motors enable the system to accept steering commands and to correct for sight-line drift. By providing an appropriate torque (computed through a properly designed control system, say) through the torque motors, the system can be processed relative to the inertial space to achieve some desired LOS.

Referring to Fig. 1, the LOS stabilization system consists of four main modules, namely the rotor, the inner gimbal, the outer gimbal and the mirror. Having identified these modules, the following coordinate frames can be defined:

- the vehicle–system reference frame (x_0, y_0, z_0), assumed fixed;
- the outer gimbal frame (x_1, y_1, z_1);
- the inner gimbal and rotor frame (x_2, y_2, z_2);
- the mirror frame (x_3, y_3, z_3).

By defining these coordinate frames for the vehicle–system, the rotor, the inner gimbal, the outer gimbal, and the mirror, and applying the Lagrange equations of motion (Arnold 1987), a dynamical model of the LOS stabilization platform can be derived and is given as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + T(q,\dot{q},\dot{q}_3) = u \qquad (3.1)$$

where $q = [q_1, q_2]^T$ is a vector of angular displacements about axes 1 and 2, $u = [u_1, u_2]^T$ is the torque vector acting on axes 1 and 2, and $\dot{q}_3$ is the angular velocity of the flywheel (set as a constant according to the user's operational requirement). By using the notations $s_i = \sin q_i$ and $c_i = \cos q_i$, we can write the inertia matrix $M(q)$ as

$$M(q) = \begin{bmatrix} M_{11}(q) & 0 \\ 0 & M_{22}(q) \end{bmatrix},$$

where

$$M_{11}(q) = m_1 + m_4 + (m_2 - m_4 + m_8)c_2^2 + \tfrac{1}{2}(m_5 + m_7) + \tfrac{1}{2}(m_7 - m_5)s_2,$$

$$M_{22}(q) = m_3 + \tfrac{1}{4}m_6 + m_8.$$

The matrix $C(q,\dot{q})$ is defined by the Christoffel symbols (Slotine and Li 1991) as

$$C(q,\dot{q}) = \begin{bmatrix} C_{11}(q,\dot{q}) & C_{12}(q,\dot{q}) \\ C_{21}(q,\dot{q}) & 0 \end{bmatrix},$$

where

$$C_{11}(q,\dot{q}) = -(m_2 - m_4 + m_8)\dot{q}_2 s_2 c_2 + \tfrac{1}{4}(m_7 - m_5)\dot{q}_2 c_2,$$

$$C_{12}(q,\dot{q}) = -(m_2 - m_4 + m_8)\dot{q}_1 s_2 c_2 + \tfrac{1}{4}(m_7 - m_5)\dot{q}_1 c_2 + m_9 \dot{q}_1 s_2 c_2,$$

$$C_{21}(q,\dot{q}) = (m_2 - m_4 + m_8)\dot{q}_1 s_2 c_2 - \tfrac{1}{4}(m_7 - m_5)\dot{q}_1 c_2 - m_9 \dot{q}_1 s_2 c_2.$$

The coupling term associated with the flywheel is

$$T(q,\dot{q},\dot{q}_3) = \begin{bmatrix} m_9 \dot{q}_3 \dot{q}_2 c_2 \\ -m_9 \dot{q}_3 \dot{q}_1 c_2 \end{bmatrix}$$

and, in all the above expressions, m_i is the moment of inertia of each element along the principal axis of its own frame, and these are defined as follows:

- m_1, moment of inertia of outer gimbal about $x_1 = x_0$;

- m_2, m_3, m_4, moments of inertia of inner gimbal about x_2, y_2, z_2 respectively;
- m_5, m_6, m_7, moments of inertia of inner gimbal about x_3, y_3, z_3 respectively;
- m_8, moments of inertia of rotor (flywheel) about x_2;
- m_9, moments of inertia of rotor (flywheel) about z_2.

Clearly, from (3.1) it can be seen that the gyro system is a highly nonlinear system and there are cross-couplings between axes 1 and 2. The components of the cross-couplings acting through the term T are proportional to $\dot{q}_3$ and arise from the flywheel kinetic energy term (Loh 1991, Lee *et al.* 1996). They differ from the gravitational force term G in the robot dynamic equations (Craig 1980) where $G(q)$ is a function of q only. Nevertheless, motivated by the nonlinear dynamical equation structure of robot dynamics, we shall utilize the following definitions: $P_1 = m_1 + m_4$; $P_2 = m_2 - m_4 + m_8$; $P_3 = m_5 + m_7$; $P_4 = m_3 + \frac{1}{4}m_6 + m_8$; $P_5 = m_7 - m_5$; $P_6 = m_9$. Then the dynamical equation (3.1) can be rewritten as

$$M(q)\ddot{q} + B(q, \dot{q}, \dot{q}_3)\dot{q} = u, \tag{3.2}$$

where

$$M(q) = \begin{bmatrix} P_1 + P_2c_2^2 + \frac{1}{2}P_3 + \frac{1}{2}P_5s_2 & 0 \\ 0 & P_4 \end{bmatrix} > 0,$$

and $B(q, \dot{q}, \dot{q}_3)$ is obtained by combining the two elements of T into $C_{12}(q, \dot{q})$ and $C_{21}(q, \dot{q})$ respectively as follows:

$$B(q, \dot{q}, \dot{q}_3) = \begin{bmatrix} B_{11}(q, \dot{q}) & B_{12}(q, \dot{q}, \dot{q}_3) \\ B_{21}(q, \dot{q}, \dot{q}_3) & 0 \end{bmatrix},$$

where

$$B_{11}(q, \dot{q}) = -P_2\dot{q}_2s_2c_2 + \tfrac{1}{4}P_5\dot{q}_2c_2$$

$$B_{12}(q, \dot{q}, \dot{q}_3) = -P_2\dot{q}_1s_2c_2 + \tfrac{1}{4}P_5\dot{q}_1c_2 + P_6\dot{q}_1s_2c_2 + P_6\dot{q}_3c_2,$$

$$B_{21}(q, \dot{q}, \dot{q}_3) = P_2\dot{q}_1s_2c_2 - \tfrac{1}{4}P_5\dot{q}_1c_2 - P_6\dot{q}_1s_2c_2 - P_6\dot{q}_3c_2.$$

Note that these terms are purposefully combined in such a manner so as to yield $Z = \dot{M} - 2B$ as a skew-symmetric matrix; this will subsequently assure that the methods developed previously by Lee and Tan (1993), Ge and Lee (1997) and Ge *et al.* (1997) can be modified and extended for application to the gyro-mirror platform. With this arrangement, the elements of the matrix Z, in fact, are

$$Z_{11} = Z_{22} = 0,$$

$$Z_{12} = 2P_2\dot{q}_1s_2c_2 - \tfrac{1}{2}P_5\dot{q}_1c_2 - 2P_6\dot{q}_1s_2c_2 - 2P_6\dot{q}_3c_2,$$

$$Z_{21} = -2P_2\dot{q}_1s_2c_2 + \tfrac{1}{2}P_5\dot{q}_1c_2 + 2P_6\dot{q}_1s_2c_2 + 2P_6\dot{q}_3c_2.$$

Thus clearly we have $Z = -Z^{\mathrm{T}}$, and hence Z is skew symmetric. The equivalent description of the gyro-mirror platform given by (3.2) and the properties indicated above will play an important role in the design and development of the real-time neural-network-based controller for the platform.

4. Neural-network-based modelling and control of the gyro-mirror platform

4.1. *Neural network-based modelling*

The dynamics of the two-degree-of-freedom gyro are described in preceding section and are given by

$$M(q)\ddot{q} + B(q, \dot{q}, \dot{q}_3)\dot{q} = u. \tag{4.1}$$

Since $M(q)$ is a function of q only, we shall take the approach that $m_{kj}(q)$, the kjth element of $M(q)$, can be approximated by a static neural network $\theta_{kj}^{\mathrm{T}}\xi_{kj}(q)$ as

$$m_{kj}(q) = \theta_{kj}^{\mathrm{T}}\xi_{kj}(q) + \epsilon_{kj}, \tag{4.2}$$

where θ_{kj} is the weight vector, $\xi_{kj}(q)$ is the neural network RBF vector and ϵ_{kj} is the approximation error. Then referring to appendix A, we have

$$M(q) = [\{\Theta\}^{\mathrm{T}} \bullet \{\mathcal{E}\}(q)] + E_{\mathrm{M}}, \tag{4.3}$$

where E_{M} is the modelling error matrix with ϵ_{kj} being the kjth element; $\{\Theta\}$ and $\{\mathcal{E}\}(q)$ are the weight and neural network RBF Ge–Lee (GL) matrices respectively, with θ_{kj} being the kjth element of $\{\Theta\}$ and $\xi_{kj}(q)$ being the kjth element of $\{\mathcal{E}\}(q)$; $\bullet$ is the GL operator. (For detailed descriptions of the notation used for the GL operator and matrix, see appendix A.)

Similarly, because $B(q, \dot{q}, \dot{q}_3)$ is a function of q, $\dot{q}$ and $\dot{q}_3$, its kjth element, $b_{kj}(q, \dot{q}, \dot{q}_3)$ can be approximated by a dynamic neural network $\alpha_{kj}^{\mathrm{T}}\xi_{kj}(x)$ as

$$b_{kj}(q, \dot{q}, \dot{q}_3) = \alpha_{kj}^{\mathrm{T}}\xi_{kj}(x) + \epsilon'_{kj} \tag{4.4}$$

where $x = [q^{\mathrm{T}}, \dot{q}^{\mathrm{T}}, \dot{q}_3]^{\mathrm{T}}$, α_{kj} is the weight vector, $\xi_{kj}(x)$ is the neural network RBF vector and ϵ'_{kj} is the corresponding approximation error. Referring to appendix A, we obtain

$$B(q, \dot{q}, \dot{q}_3) = [\{\mathcal{A}\}^{\mathrm{T}} \bullet \{\mathcal{Z}\}(x)] + E_{\mathrm{B}}, \tag{4.5}$$

where E_{B} is the modelling error matrix with ϵ'_{kj} being the kjth element, and $\{\mathcal{A}\}$ and $\{\mathcal{Z}\}(x)$ are the weight and neural network RBF GL matrices respectively, with α_{kj} being the kjth element of $\{\mathcal{A}\}$ and $\xi_{kj}(x)$ being he kjth element of $\{\mathcal{Z}\}(x)$. Thus (4.1) can be written as

$$\begin{aligned} M(q)\ddot{q} + B(q, \dot{q}, \dot{q}_3)\dot{q} &= [\{\Theta\}^{\mathrm{T}} \bullet \{\mathcal{E}\}(q)]\ddot{q} + [\{\mathcal{A}\}^{\mathrm{T}} \bullet \{\mathcal{Z}\}(x)]\dot{q} \\ &\quad + E_{\mathrm{M}}\ddot{q} + E_{\mathrm{B}}\dot{q} = u. \end{aligned} \tag{4.6}$$

As discussed in section 2, the Gaussian and the Hardy multiquadric RBF neural networks will be used for the nonlinear system modelling in the work described here.

4.2. *Controller design*

The dynamic equation of the gyro has been introduced in section 3. In this section, we shall consider how to develop a neural-network-based controller for the gyro-mirror platform. This will be based on modifications and extensions of the work by Lee and Tan (1993) and Ge *et al.* (1997). Thus let q_d, $\dot{q}_d$, $\ddot{q}_d \in R^2$ be the respective desired position, velocity and acceleration vectors. Let Λ be a positive-definite diagonal matrix and define the following quantities:

$$\epsilon = q_d - q, \tag{4.7}$$

$$\dot{q}_r = \dot{q}_d + \Lambda\epsilon, \tag{4.8}$$

$$r = \dot{q}_r - \dot{q} = \dot{\epsilon} + \Lambda\epsilon. \tag{4.9}$$

Thus, as in the work of Slotine and Li (1991), r can be regarded as a new tracking error which has the following properties.

(1) r is bounded $\Rightarrow$ ϵ and $\dot{\epsilon}$ are bounded.

(2) $r \to 0 \Rightarrow \epsilon, \dot{\epsilon} \to 0$ as $t \to \infty$.

Therefore, the stability properties of ϵ and $\dot{\epsilon}$ can be investigated by studying those of r.

At this point, note that, using (4.3) and (4.5)–(4.9), the following equation can be obtained:

$$\begin{aligned} u &= M(q)\ddot{q} + B(q, \dot{q}, \dot{q}_3)\dot{q} \\ &= M(q)\ddot{q}_r + B(q, \dot{q}, \dot{q}_3)\dot{q}_r - M(q)\dot{r} - B(q, \dot{q}, \dot{q}_3)r \\ &= [\{\Theta\}^T \bullet \{\mathcal{E}\}]\ddot{q}_r + [\{\mathcal{A}\}^T \bullet \{\mathcal{Z}\}]\dot{q}_r - M(q)\dot{r} \\ &\quad - B(q, \dot{q}, \dot{q}_3)r + E, \end{aligned} \tag{4.10}$$

where

$$E = E_M\ddot{q}_r + E_B\dot{q}_r. \tag{4.11}$$

As in the work of Slotine and Li (1991), a sliding mode control term will be incorporated into the controller to handle the modelling error and assure the closed-loop stability of the system. Thus, motivated by the neural network-based controllers previously developed by Lee *et al.* (1992), Lee and Tan (1993) and Ge *et al.* (1997), the controller for the gyro-mirror platform can take the following form:

$$u = \hat{M}_{nn}(q)\ddot{q}_r + \hat{B}_{nn}(x)\dot{q}_r + Kr + K_i\int_0^t r\,d\tau + K_s\,\mathrm{sgn}(r), \tag{4.12}$$

where $\hat{M}_{nn}(q) = \{\hat{\Theta}\}^T \bullet \{\mathcal{E}\}(q)$ and $\hat{B}_{nn}(x) = \{\mathcal{A}\}^T \bullet \{\mathcal{Z}\}(x)$, with $\{\hat{\Theta}\}$ and $\{\hat{\mathcal{A}}\}$ being the estimates of $\{\Theta\}$ and $\{\mathcal{A}\}$ respectively. Also the gains are chosen so that $K > 0$, $K_i \geq 0$ and $K_s > \|E\|$.

In the design of the neural-network-based controller described above, it is pertinent to point out that the neural networks are only used to compute $\hat{M}_{nn}(q)$ and $\hat{B}_{nn}(x)$. The inputs to the networks are q, $\dot{q}$ and $\dot{q}_3$, and no matrix inversion is needed. Further, by combining (4.10) and (4.12), the following error equation can be derived:

$$\begin{aligned} &M(q)\dot{r} + B(x)r + Kr + K_i\int_0^t r(\tau)\,d\tau + K_s\,\mathrm{sgn}(r) \\ &\quad = [\{\tilde{\Theta}\}^T \bullet \{\mathcal{E}\}(q)]\ddot{q}_r + [\{\tilde{\mathcal{A}}\}^T \bullet \{\mathcal{Z}\}(x)]\dot{q}_r + E. \end{aligned} \tag{4.13}$$

Based on this error equation, the properties of the neural-network-based controller applied to the gyro-mirror platform can be described as follows.

Proposition 1: *For the neural-network-based control law given by* (4.12), *the closed-loop system given by* (4.13) *is asymptotically stable, that is* $r \to 0$ *as* $t \to \infty$, *under the following adaptive laws:*

$$\dot{\hat{\theta}}_i = \Gamma_i \bullet \{\xi_i(q)\}\ddot{q}_r r_i, \tag{4.14}$$

$$\dot{\hat{\alpha}}_i = Q_i \bullet \{\xi_i(x)\}\dot{q}_r r_i, \tag{4.15}$$

where $\Gamma_i = \Gamma_i^T > 0$, $Q_i = Q_i^T > 0$, *and* $\hat{\theta}_i$ *and* $\hat{\alpha}_i$ *are elements of* $\{\hat{\Theta}\}$ *and* $\{\hat{\mathcal{A}}\}$ *respectively.*

It can further be shown that for the overall system, we also have $\hat{\theta}_i, \hat{\alpha}_i \in L^\infty$ *and* $\epsilon \in L_n^2 \cap L_n^\infty$, ϵ *is continuous,* ϵ *and* $\dot{\epsilon} \to 0$ *as* $t \to \infty$.

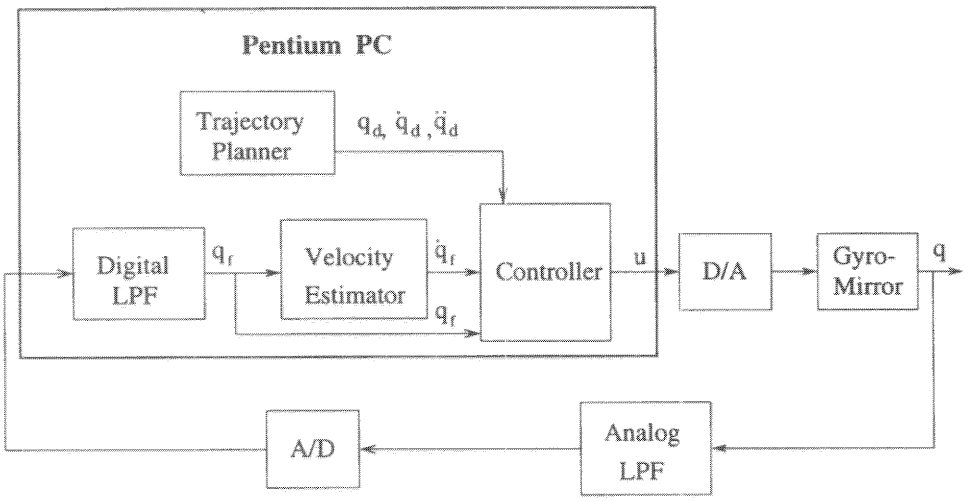
The detailed proof of proposition 1 is given in appendix B. The result above essentially indicates that the neural-network-based controller developed in this work will provide stable control of the gyro-mirror platform and will achieve asymptotic tracking of the desired reference signals. The actual performance attained will be investigated in the next section with real-time experimental trials.

5. Real-time experimental results

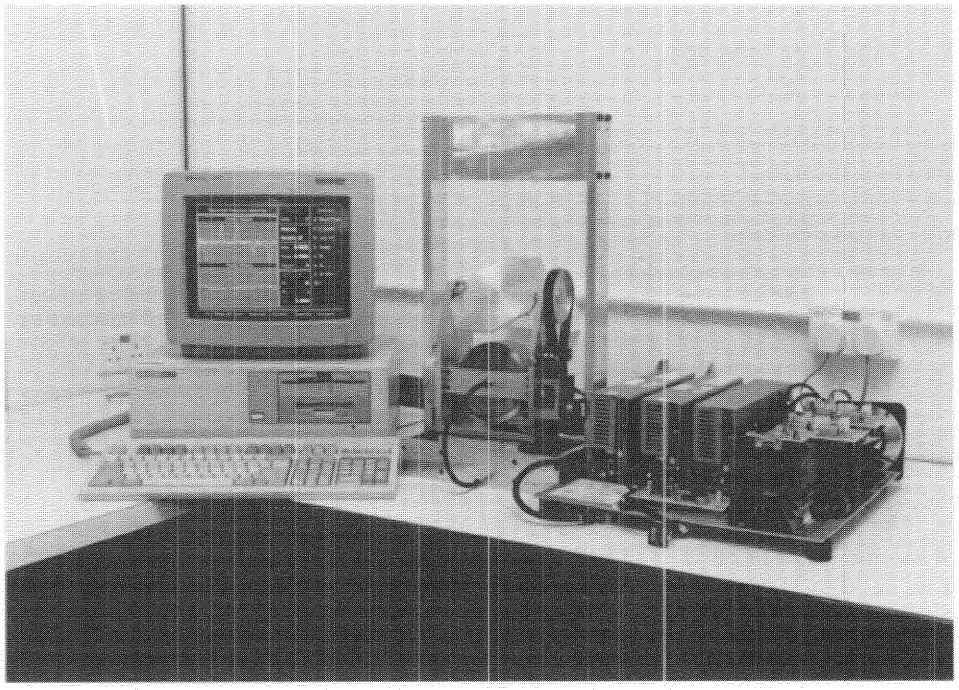
5.1. *Experimental set-up*

The dynamics of the gyro-mirror LOS stabilization platform have been previously described. In this section, real-time experimental results will be presented to demonstrate the effectiveness of the proposed neural-network-based controller. These experiments also serve to verify the analytical results in a proptotype real-time application.

The set-up for the real-time experiments is shown in the block diagram in Fig. 2(*a*) and in the photograph in Fig. 2(*b*). The angular positions of the two axes are measured by two potentiometers fixed to the two gimbals. Since the plant does not have tachometers for the required velocity measurements (in a typical application, there is only sufficient space to mount either a potentiometer or a tachometer, but not both), a *Lagrange interpolation polynomial* is used to construct the velocity signal from the measured position signal. An analogue



(*a*)



(*b*)

Figure 2. (*a*) Block diagram of the experimental set-up; LPF, low-pass filter; D/A, digital-to-analogue; A/D, analogue to digital. (*b*) Photograph of the experimental set-up.

low-pass filter is used to attentuate the noise in the position signal before the signal is connected to the analogue-to-digital card. After filtering, the noise content is actually rather low and will be near the level of quantization noise. However, this still does not meet the requirement for velocity estimation, because the reconstruction of the velocity signal is effectively equivalent to a procedure of taking first derivatives. Therefore, a software digital filter is also used to reduce the quantization noise. For this, a second-order low-pass digital filter is ultilized:

$$q_{fi}(kh) = a[q_i(kh) + 2q_i(kh-h) + q_i(kh-2h)] - bq_{fi}(kh-h) - cq_{fi}(kh-2h). \quad (5.1)$$

The filtered position variable q_f is then used for the velocity estimation. A two-degree *Langrange interpolation polynomial* is used here:

$$\dot{q}_{fi}(kh) = \frac{q_{fi}(kh-2h) - 4q_{fi}(kh-h) + 3q_{fi}(kh)}{2h}. \quad (5.2)$$

The estimated velocity $\dot{q}_f$ together with the signals q_f, q_d, $\dot{q}_d$ and $\ddot{q}_d$ are the inputs to the controller, and u is the output of the controller. The computed control signal u is sent via the digital-to-analogue converters to the power amplifiers of the two torque motors to drive the gyro-mirror platform. For the experiments considered here, a periodic sinusoid

$$q_d(t) = \begin{bmatrix} 0.1 \sin(0.2\pi t) \\ 0.2 \sin(0.2\pi) \end{bmatrix}$$

is utilized as the desired trajectory for position tracking.

5.2. *Experimental results*

As mentioned above, the real-time experiments serve to verify the analytical results in a prototype real-time application. The performance of the proposed neural-network-based controller is investigated here and also compared with a basic proportional–integral–differential (PID) controller.

For the proposed neural-network-based controller, 30-node RBF neural networks were implemented for the elements of $\hat{M}_{nn}(q)$ and 60-node neural networks were chosen for the elements of $\hat{B}_{nn}(x)$. The sampling period was set to be 10 ms and the speed of the flywheel was set at $\dot{q}_3 = 209.0\,\text{rad s}^{-1}$. Owing to the highly nonlinear nature of the gyro-mirror platform dynamics, it is also essential to note that in a typical application the assumption must be made that the nonlinear dynamic parameters of the platform are not known to any reliable degree of accuracy, and thus the initial values of the GL parameter matrices $\{\hat{\Theta}\}(0)$ and $\{\hat{A}\}(0)$ of the neural networks were realistically initialized with zero values for the real-time experiments.

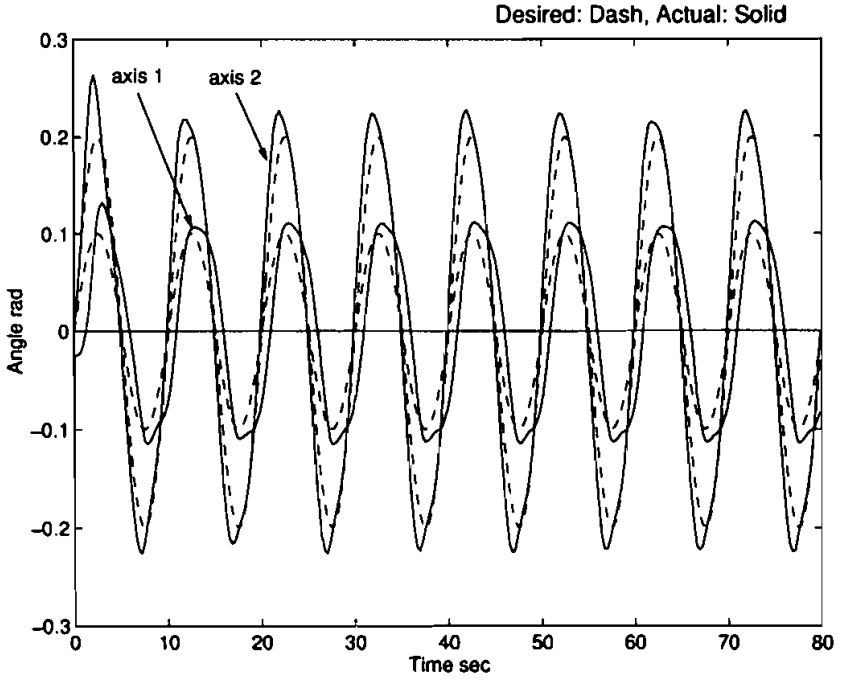
5.2.1. *Proportional–integral–differential-type controller.* For comparison, we first investigate the position control performance when a PID-type controller is used. The PID-type controller can be easily obtained from (4.12) by setting $K_s = 0$ and the adaptation gains $Q_i = 0$, $\Gamma_i = 0$, that is

$$u = Kr + K_i \int_0^t r \, d\tau.$$

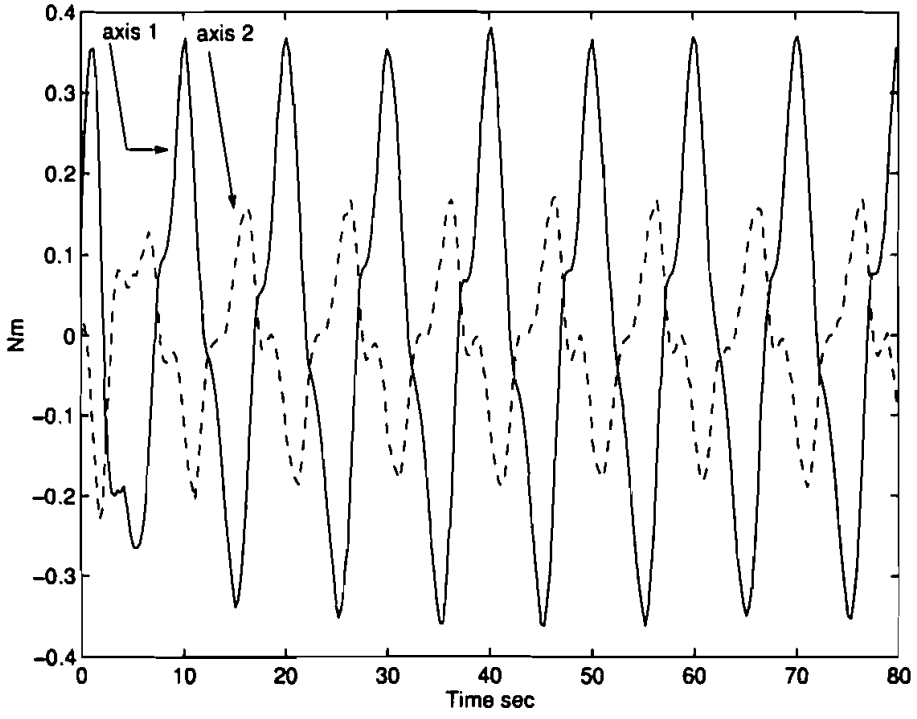
Clearly, it is a PID-type controller because of the definition of r given in (4.9). Note that, as mentioned above, in a typical application environment, the dynamic parameters of the gyro-mirror platform are not known accurately, and therefore only a very conservative tuning of the PID-type controller may be attempted. By trial and error the set of control parameters are thus chosen as

$$K = \text{diag}[0.1], \quad K_i = 0.1, \quad \Lambda = \text{diag}[35.0].$$

For this PID-type controller, the position tracking performance and control signals are shown in Figs 3(*a*) and (*b*) respectively. It can be seen from these figures that some tracking errors exist. Actually, the main problem here is not just in the tracking errors; the simple PID controller is also unable to respond effectively in one axis without significant undesirable interaction in the other axis (caused by the coupled nonlinear dynamics of the platform).



3(*a*)



(*b*)

Figure 3. (*a*) Position tracking performance and (*b*) control signals of the PID controller.

5.2.2. *Radial basis function neural-network-based controller.* For the hardware under study, the values of angular displacements q_1, q_2 are bounded by $[-1.0, 1.0]$ rad and angular velocities $\dot{q}_1$, $\dot{q}_2$ by $[-0.24, 0.24]\,\text{rad s}^{-1}$. Therefore, the two-dimensional input space for $\hat{M}_{mn}(q)$ is defined as $S_1 \in (q_1, q_2 \in [-1.0, 1.0])$ and the space is divided into square grid with the centres c_i placed at the crossings of the grid as shown in Fig. 4 with σ_i being its variance and d the distance between two adjacent nodes. For $\hat{B}_{mn}(x)$, the four-dimensional input space is defined as $S_2 \in (q_1, q_2 \in [-1.0, 1.0]$, $\dot{q}_1$, $\dot{q}_2 \in [-0.24, 0.24])$ and the space is also divided into grids with centres also at the crossings. The node number can be determined by the distance d. Smaller d means finer grids and more nodes.

To investigate the robustness and effectiveness of the proposed neural network controllers, the robust sliding mode term $K_s \operatorname{sgn}(r)$ in the neural network controller (4.12) is removed by setting $K_s = 0$.

For the Gaussian RBF neural-network-based controller, the adaptive mechanisms are activated with gains $Q_i = \text{diag}[0.0002]$ and $\Gamma_i = \text{diag}[0.05]$. The position tracking performance and control signals are shown in Figs 5(*a*) and (*b*), respectively. For the Hardy multiquadric RBF neural-networks-based controller, the corresponding adaptive mechanisms are activated with gains $Q_i = \text{diag}[0.0002]$ and $\Gamma_i = \text{diag}[0.05]$. The position tracking performance and control signals are shown in Figs 6(*a*) and (*b*) respectively. The variations in $\|\hat{M}\|_2$ and $\|\hat{B}\|_2$ are shown in Figs 7(*a*) and (*b*), respectively. It may be noted that, in both cases, uniformly stable operation is achieved, the tracking errors are kept small, and the gyro-mirror platform operates with no appreciable interaction in the two axes. Although theoretically the system is stable as long as the adaptation

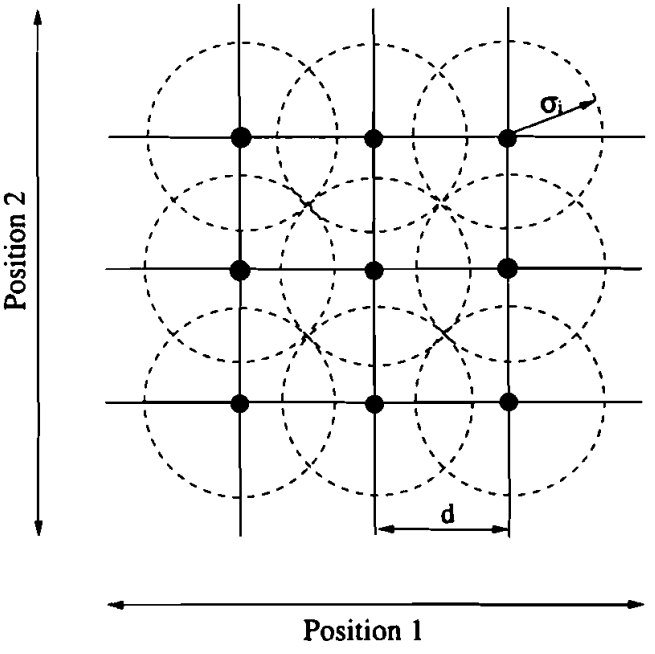
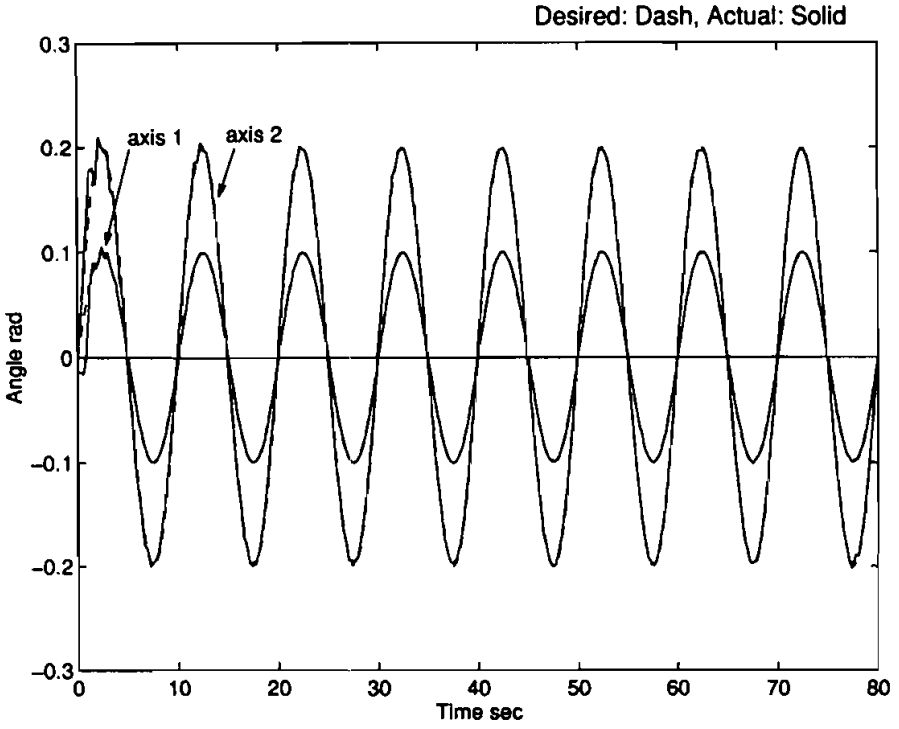
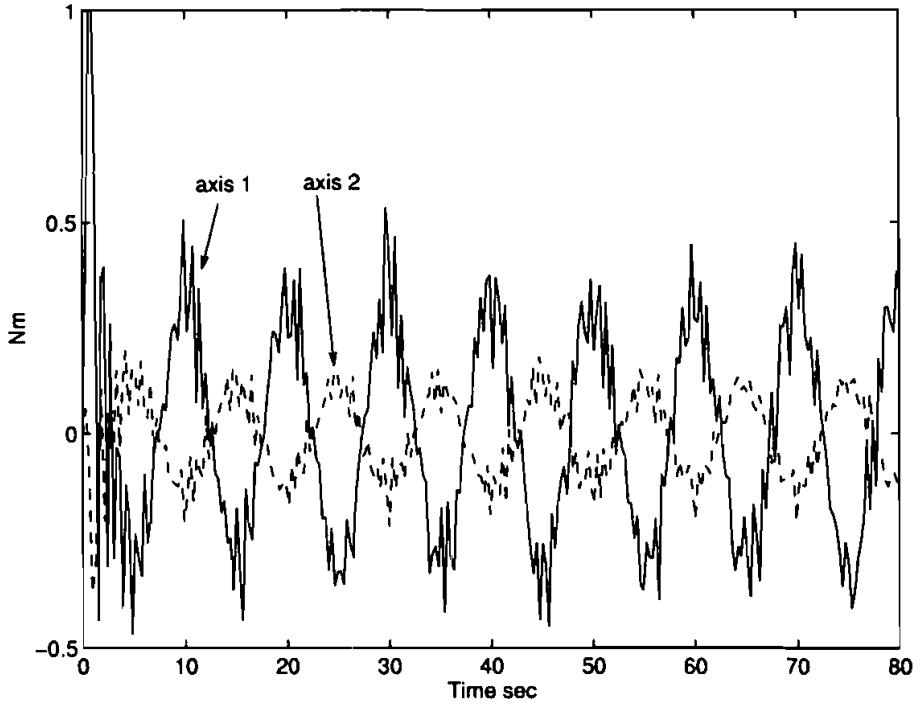


Figure 4. Two-dimensional input space divided by grids.



(*a*)

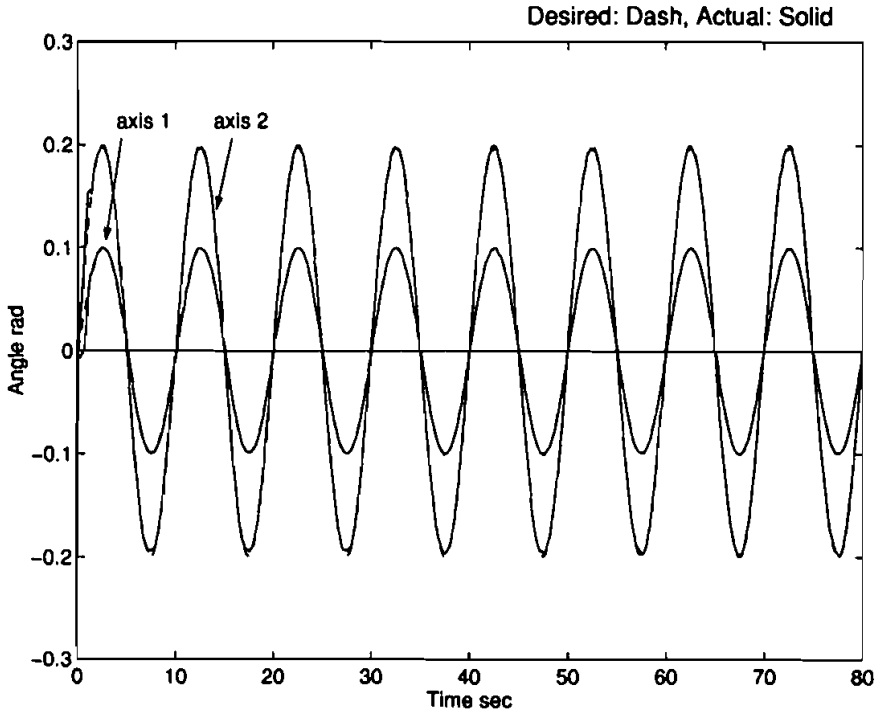


(*b*)

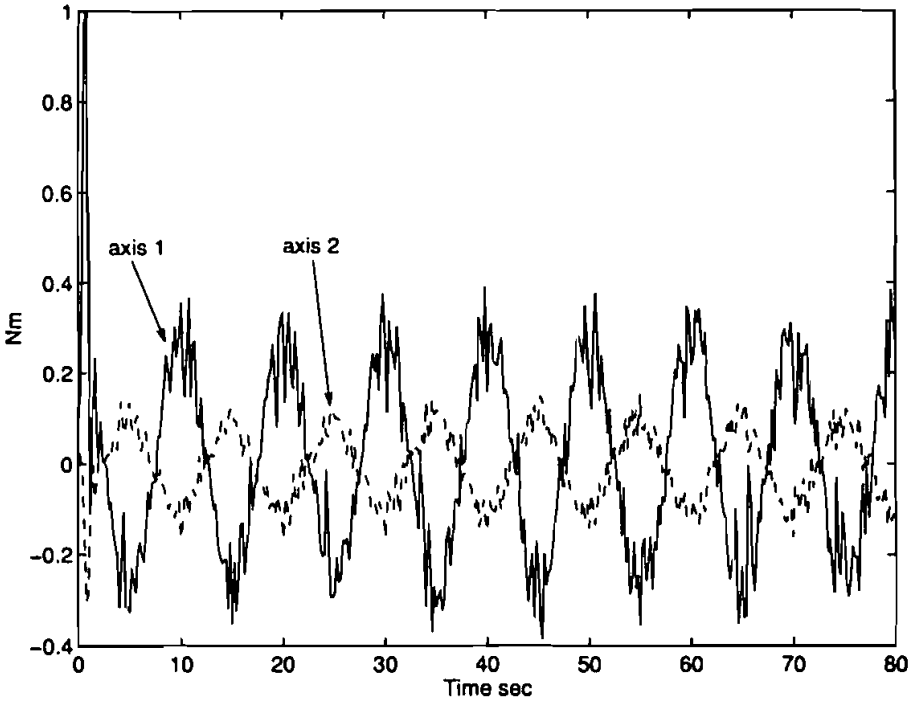
Figure 5. (*a*) Position tracking performance and (*b*) control signals of the Gaussian RBF controller.

gains $\Gamma_i^{\mathrm{T}} = \Gamma_i > 0$ and $Q_i^{\mathrm{T}} = Q_i > 0$, they cannot be too large nor too small because of the discrete implementation of the controller.

Until now, there is no theoretical result to show which is more efficient in function approximation. Through comprehensive comparison studies for this particular system, we found that firstly the controller based on the Hardy multiquadric RBF neural networks performs slightly better than that of Gaussian RBF neural networks and secondly both of the two neural network controllers give satisfactory performance when the number of nodes is in the range 20–100. When the



(a)



(b)

Figure 6. (*a*) Position tracking performance and (*b*) control signals of the Hardy multiquadric RBF controller.

number of nodes is too small, the performance actually deteriorates because the neural networks are not large enough to approximate the nonlinear functions. If the size of neural networks becomes larger, while the increase in control performance is not very significant, the computational overhead may render the system unstable.

Note that it is undesirable to implement directly the sliding control term to cancel the effect of the neural network approximation errors because of the chattering caused. It can be easily solved by introducing a boundary layer in practical realization as discussed by Xu *et al.* (1989) and Slotine and Li (1991).

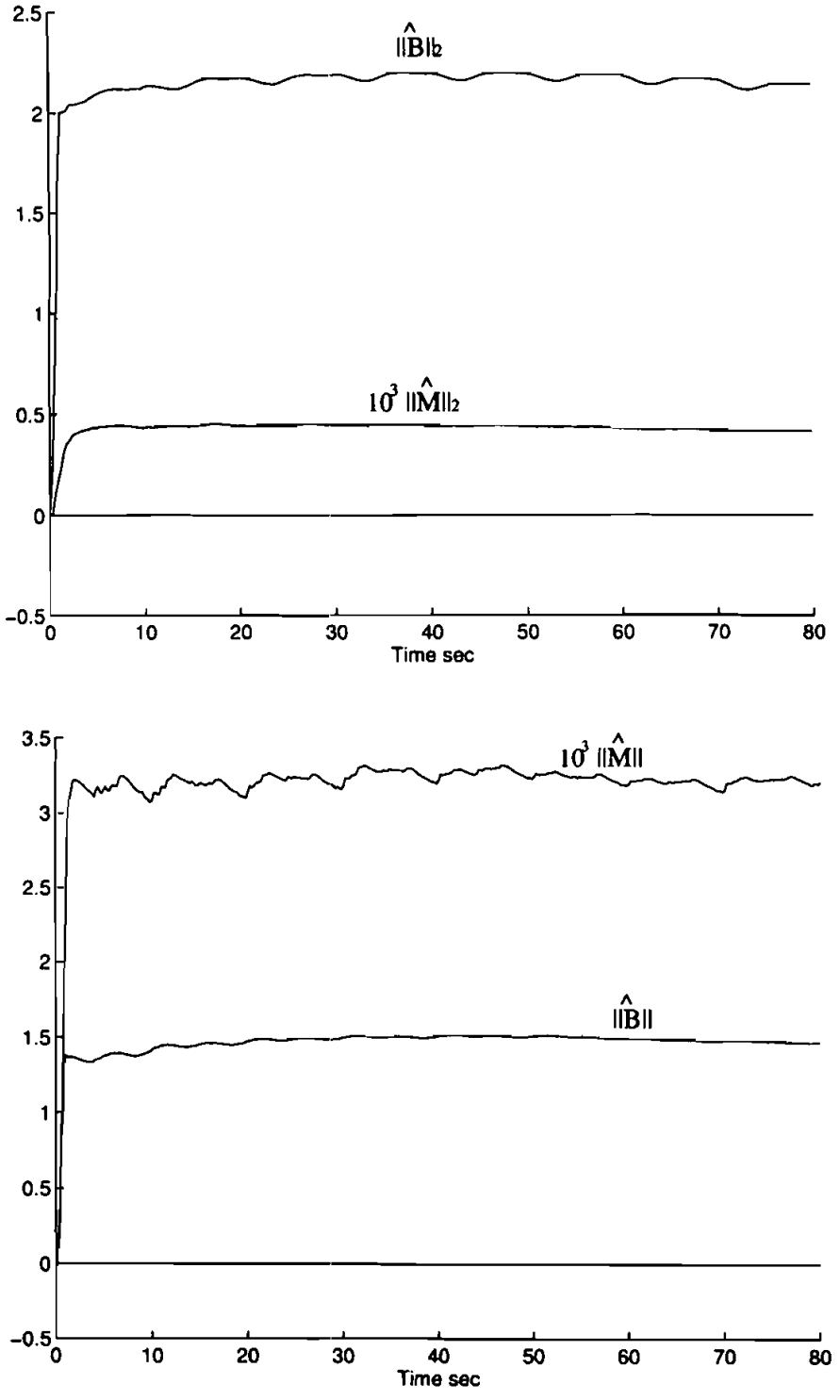


Figure 7. $\|\hat{M}\|_2$ and $\|\hat{B}\|_2$ (the Hardy multiquadric RBF controller).

6. Conclusion

In this paper, we have considered the real-time neural network-based control of a gyro-mirror LOS stabilization platform. The appropriate neural-network-based modelling and controller design have been discussed and, in the paper, real-time experimental results in applying the proposed controller to a pilot-scale gyro-mirror platform are presented to demonstrate its effectiveness. These experiments also serve to verify the analytical results in a prototype real-time application. Current work is directed at in-depth comparative studies for the neural-network-based control system involving different classes of

RBFs, and this is aimed at a better understanding of their relative advantages under various operating conditions involving different distributions of available data points, and the complexity of implementation in practical embedded systems.

Appendix A. Ge–Lee product operator •

The definition of the Ge–Lee (GL) vectors, denoted by $\{*\}$ and the GL product operator • was introduced by Ge and Lee (1997) and Ge *et al.* (1997). To improve the readability of this paper, we discuss it briefly here.

Define I_0 as the set of integers, and $\theta_{ij}, \xi_{ij} \in R^{n_{ij}}$, $n_{ij} \in I_0, i = 1, \ldots, n, j = 1, \ldots, n$. In this paper, θ_{ij} is taken as the weight vector and ξ_{ij} is the network basis function vector. Then define θ_i and θ_i^{T} as follows:

$$\theta_i = \begin{bmatrix} \theta_{i1} \\ \theta_{i2} \\ \vdots \\ \theta_{ik} \end{bmatrix}, \quad \theta_i^{\mathrm{T}} = [\theta_{i1}^{\mathrm{T}} \ \theta_{i2}^{\mathrm{T}} \ \ldots \ \theta_{ik}^{\mathrm{T}}].$$

Further, let

$$\{\theta_i\} = \{\theta_{i1}, \theta_{i2}, \ldots, \theta_{in}\}, \tag{A.1}$$

$$\{\theta_i\}^{\mathrm{T}} = \{\theta_{i1}^{\mathrm{T}}, \theta_{i2}^{\mathrm{T}}, \ldots, \theta_{in}^{\mathrm{T}}\}, \tag{A.2}$$

$$\{\xi_i\} = \{\xi_{i1}, \xi_{i2}, \ldots, \xi_{in}\}. \tag{A.3}$$

Then the GL product is defined as

$$[\{\theta_i\}^{\mathrm{T}} \bullet \{\xi_i\}] = [\theta_{i1}^{\mathrm{T}}\xi_{i1}, \theta_{i2}^{\mathrm{T}}\xi_{i2}, \ldots, \theta_{in}^{\mathrm{T}}\xi_{in}]. \tag{A.4}$$

By stacking $[\{\theta_i\}^{\mathrm{T}} \bullet \{\xi_i\}]$ in a column, we have the corresponding GL matrix product:

$$[\{\Theta\}^{\mathrm{T}} \bullet \{\mathcal{E}\}] := \begin{bmatrix} \{\theta_1\}^{\mathrm{T}} \bullet \{\xi_1\} \\ \{\theta_2\}^{\mathrm{T}} \bullet \{\xi_2\} \\ \vdots \\ \{\theta_n\}^{\mathrm{T}} \bullet \{\xi_n\} \end{bmatrix} \tag{A.5}$$

where

$$\{\Theta\} = \begin{Bmatrix} \theta_{11} & \theta_{12} & \cdots & \theta_{1n} \\ \theta_{21} & \theta_{22} & \cdots & \theta_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \theta_{n1} & \theta_{n2} & \cdots & \theta_{nn} \end{Bmatrix} = \begin{Bmatrix} \{\theta_1\} \\ \{\theta_2\} \\ \vdots \\ \{\theta_n\} \end{Bmatrix}, \tag{A.6}$$

$$\{\mathcal{E}\} = \begin{Bmatrix} \xi_{11} & \xi_{12} & \cdots & \xi_{1n} \\ \xi_{21} & \xi_{22} & \cdots & \xi_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \xi_{n1} & \xi_{n2} & \cdots & \xi_{nn} \end{Bmatrix} = \begin{Bmatrix} \{\xi_1\} \\ \{\xi_2\} \\ \vdots \\ \{\xi_n\} \end{Bmatrix}, \tag{A.7}$$

and the transpose of the GL matrix $\{\Theta\}$ is defined as

$$\{\Theta\}^{\mathrm{T}} = \begin{Bmatrix} \theta_{11}^{\mathrm{T}} & \theta_{12}^{\mathrm{T}} & \cdots & \theta_{1n}^{\mathrm{T}} \\ \theta_{21}^{\mathrm{T}} & \theta_{22}^{\mathrm{T}} & \cdots & \theta_{2n}^{\mathrm{T}} \\ \vdots & \vdots & \vdots & \vdots \\ \theta_{n1}^{\mathrm{T}} & \theta_{n2}^{\mathrm{T}} & \cdots & \theta_{nn}^{\mathrm{T}} \end{Bmatrix}. \tag{A.8}$$

In the paper, $[*]$ is used to denote an ordinary matrix, and $\{*\}$ for a GL matrix.

Appendix B. Proof of asymptotic stability

The proof is obtained by extending the stability proof given by Ge *et al.* (1997). First select the non-negative function V as

$$V = \tfrac{1}{2}\left[r^{\mathrm{T}}Mr + \left(\int_0^t r\,\mathrm{d}\tau\right)^{\mathrm{T}} K_i \left(\int_0^t r\,\mathrm{d}\tau\right) + \sum_{i=1}^{n} \tilde{\theta}_i^{\mathrm{T}} \Gamma_i^{-1} \tilde{\theta}_i + \sum_{i=1}^{n} \tilde{\alpha}_i^{\mathrm{T}} Q_i^{-1} \tilde{\alpha}_i\right], \tag{B.1}$$

where Γ_i *and* Q_i are two-dimensional compatible symmetric positive-definite matrices.

Then the time derivative of V is as follows:

$$\dot{V} = r^{\mathrm{T}}\left[M(q)\dot{r} + \tfrac{1}{2}\dot{M}(q)r + K_i\left(\int_0^t r\,\mathrm{d}\tau\right)\right] + \sum_{i=1}^{n} \tilde{\theta}_i^{\mathrm{T}} \Gamma_i^{-1} \dot{\tilde{\theta}}_i + \sum_{i=1}^{n} \tilde{\alpha}_i^{\mathrm{T}} Q_i^{-1} \dot{\tilde{\alpha}}_i \tag{B.2}$$

and, because $\dot{M} - 2B$ is skew symmetric, that is $r^{\mathrm{T}}(\dot{M} - 2B)r = 0$, we have

$$\dot{V} = r^{\mathrm{T}}\left[M(q)\dot{r} + B(q, \dot{q})r + K_i\left(\int_0^t r\,\mathrm{d}\tau\right)\right] + \sum_{i=1}^{n} \tilde{\theta}_i^{\mathrm{T}} \Gamma_i^{-1} \dot{\tilde{\theta}}_i + \sum_{i=1}^{n} \tilde{\alpha}_i^{\mathrm{T}} Q_i^{-1} \dot{\tilde{\alpha}}_i. \tag{B.3}$$

From the error equation (4.13), we have

$$M(q)\dot{r} + B(x)r + K_i\left(\int_0^t r\,\mathrm{d}\tau\right) = [\{\tilde{\Theta}\}^{\mathrm{T}} \bullet \{\mathcal{E}\}(q)]\ddot{q}_r + [\{\tilde{\mathcal{A}}\}^{\mathrm{T}} \bullet \{\mathcal{Z}\}(x)]\dot{q}_r - Kr + E - K_{\mathrm{s}} \operatorname{sgn}(r) \tag{B.4}$$

and the following equation can be obtained:

$$\dot{V} = -r^{\mathrm{T}}Kr + r^{\mathrm{T}}[E - K_{\mathrm{s}} \operatorname{sgn}(r)] + r^{\mathrm{T}}[\{\tilde{\Theta}\}^{\mathrm{T}} \bullet \{\mathcal{E}\}(q)]\ddot{q}_r + r^{\mathrm{T}}[\{\tilde{\mathcal{A}}\}^{\mathrm{T}} \bullet \{\mathcal{Z}\}(x)]\dot{q}_r + \sum_{i=1}^{m} \tilde{\theta}_i^{\mathrm{T}} \Gamma_i^{-1} \dot{\tilde{\theta}}_i + \sum_{i=1}^{n} \tilde{\alpha}_i^{\mathrm{T}} Q_i^{-1} \dot{\tilde{\alpha}}_i. \tag{B.5}$$

Note that

$$r^{\mathrm{T}}[\{\tilde{\theta}\}^{\mathrm{T}} \bullet \{\mathcal{E}\}]\dot{q}_r = [r_1 \ r_2 \ \cdots \ r_n] \begin{bmatrix} \{\tilde{\theta}_1\}^{\mathrm{T}} \bullet \{\xi_1(q)\}\ddot{q}_r \\ \{\tilde{\theta}_2\}^{\mathrm{T}} \bullet \{\xi_2(q)\}\ddot{q}_r \\ \vdots \\ \{\tilde{\theta}_n\}^{\mathrm{T}} \bullet \{\xi_n(q)\}\ddot{q}_r \end{bmatrix}$$

$$= \sum_{i=1}^{n} \{\tilde{\theta}_i\}^{\mathrm{T}} \bullet \{\xi_i(q)\}\ddot{q}_r r_i \tag{B.6}$$

and, similarly, we have

$$r^{\mathrm{T}}[\{\tilde{\mathcal{A}}\}^{\mathrm{T}} \bullet \{\mathcal{Z}\}]\dot{q}_r = \sum_{i=1}^{n} \{\tilde{\alpha}_i\}^{\mathrm{T}} \bullet \{\xi_i(x)\}\dot{q}_r r_i.$$

Substituting the above equation into (B.5), we have

$$\dot{V} = -r^{\mathrm{T}}Kr + r^{\mathrm{T}}[E - K_{\mathrm{s}} \operatorname{sgn}(r)] + \sum_{i=1}^{n} \{\tilde{\theta}_i\}^{\mathrm{T}} \bullet \{\xi_i(q)\}\ddot{q}_r r_i + \sum_{i=1}^{n} \{\tilde{\alpha}_i\} \bullet \{\xi_i(x)\}\dot{q}_r r_i + \sum_{i=1}^{m} \tilde{\theta}_i^{\mathrm{T}} \Gamma_i^{-1} \dot{\tilde{\theta}}_i + \sum_{i=1}^{n} \tilde{\alpha}_i^{\mathrm{T}} Q_i^{-1} \dot{\tilde{\alpha}}_i. \tag{B.7}$$

At this point, let

$$\dot{\tilde{\theta}}_i = -\Gamma_i \bullet \{\xi_i(q)\}\ddot{q}_r r_i, \tag{B.8}$$

$$\dot{\tilde{\alpha}}_i = -Q_i \bullet \{\xi_i(x)\}\dot{q}_r r_i, \tag{B.9}$$

and, because $\dot{\tilde{\theta}}_i = -\dot{\hat{\theta}}_i, \dot{\tilde{\alpha}}_i = -\dot{\hat{\alpha}}_i$, we have

$$\dot{\hat{\theta}}_i = \Gamma_i \bullet \{\xi_i(q)\}\ddot{q}_r r_i, \tag{B.10}$$

$$\dot{\hat{\alpha}}_i = Q_i \bullet \{\xi_i(x)\}\dot{q}_r r_i. \tag{B.11}$$

Therefore $\dot{V} = -r^{\mathrm{T}}Kr + r^{\mathrm{T}}(E - K_{\mathrm{s}} \operatorname{sgn}(r)) \leq 0$.

Thus, V is a Laypunov function and

$$\lambda_{\min}(K) \int_0^t r^{\mathrm{T}} r(\tau)\, \mathrm{d}\tau \leq \int_0^t r^{\mathrm{T}} Kr(\tau)\, \mathrm{d}\tau \leq V(0). \tag{B.12}$$

The following deductions may also be made.

(*a*) Since $V(0)$ and $\lambda_{\min}(K)$ are positive constants, it follows that $r \in L_n^2$. Consequently, $\epsilon \in L_n^2 \cap L_n^\infty$, ϵ is continuous and $\epsilon \to 0$ as $t \to \infty$, and $\dot{\epsilon} \in L_n^2$ (Desoer and Vidyasagar 1975).

(*b*) Since $V \leq r^{\mathrm{T}}Kr \leq 0$, it follows that $0 \leq V(t) \leq V(0)$, $\forall t \geq 0$. Hence $V(t) \in L^\infty \Rightarrow \int_0^t r(\tau)\, \mathrm{d}\tau$, $\tilde{\theta}_i$ and $\tilde{\alpha}_i \in L^\infty$, that is $\hat{\theta}_i$ and $\hat{\alpha}_i \in L^\infty$.

Finally, by noting that $r \in L_2^n$, q_{d}, $\dot{q}_{\mathrm{d}}$ $\ddot{q}_{\mathrm{d}} \in L^\infty$, and $\{\mathcal{E}\}$, $\{\mathcal{Z}\}$ are of bounded basis functions, we obtain that $\dot{r} \in L_\infty^n$ from (4.13). Since $\dot{r} \in L_\infty^n$, r is uniformly continuous. Thus we have the following implication.

- r is uniformly continuous, and $r \in L_2^n$, $\Rightarrow r \to 0$ at $t \to \infty$, $\Rightarrow \dot{\epsilon} \to 0$.

This completes the proof. □

References

ALFORD, D. W., 1987, The development of a directional gyroscope for remotely piloted vehicles and similar applications. *Proceedings of the Institution of Mechanical Engineers Conference on Mechanical Technology of Inertial Devices*, Newcastle, Australia, April 1987, pp. 1–8.

ARNOLD, V. I., 1978, *Mathematical Methods of Classical Mechanics* (New York: Springer-Verlag).

BIGLEY, W. J., and RIZZO, V. J., 1987, Wideband linear quadratic control of a gyro-stabilized electro-optical sight system. *IEEE Control Systems Magazine*, **7**, 20–24.

BROOMHEAD, D. S., and LOWE, D., 1988, Multivariable functional interpolation and adaptive networks. *Complex Systems*, **2**, 321–355.

CRAIG, J., 1980, *Introduction to Robotics: Mechanics and Control* (Reading, Massachusetts: Addison-Wesley).

DESOER, C. A., and VIDYASAGAR, M., 1975, *Feedback Systems: Input–Output Properties* (New York: Academic Press).

FRANKE, R., 1982, Scattered data interpolation: tests of some methods. *Mathematics of Computation*, **38**, 181–200.

GE, S. S., and LEE, T. H., 1997, Robust adaptive neural network control of a class of nonlinear systems. *Journal of System and Control Engineering*, **211**, 171–181.

GE, S. S., LEE, T. H., and ZHAO, Q., 1997, Real-time neural network control of a free-gyro stabilized mirror system. *Proceedings of the American Control Conference*, Albuquerque, New Mexico, U.S.A., pp. 1076–1080.

HASHIMOTO, H., 1987, Variable structure strategy for motion control systems. *Proceedings of IEEE Industrial Electronics Conference 1987*, Massachusetts, U.S.A., pp. 159–165.

KHANNA, T., 1990, *Foundations of Neural Networks* (Reading, Massachusetts: Addison-Wesley).

LEE, T. H., HANG, C. C., LIAN, L. L., and LIM, B. C., 1992, An approach to inverse nonlinear control using neural networks. *Mechatronics*, **2**, 595–611.

LEE, T. H., KOH, E. K., and LOH, M. K., 1996, Stable adaptive control of multivariable servomechanisms, with real-time application to a passive line-of-sight stabilization system. *IEEE Transactions on Industrial Electronics*, **43**, 98–105.

LEE, T. H., and TAN, W. K., 1993, Real-time parallel adaptive neural network control for nonlinear servomechanisms—an approach using direct adaptive techniques. *Mechatronics*, **3**, 705–725.

LOH, M. K., 1991, Design, development and control of a LOS stabilization system. MEng Thesis, Department of Electrical Engineering, National University of Singapore.

MICCHELLI, C. A., 1986, Interpolation of scattered data: distance matrices and conditionally ositive definite functions. *Constr. Approx.*, **2**, 11–22.

NGUYEN, D. H., and WIDROW, B., 1990, Neural networks for self-learning control systems. *IEEE Control Systems Magazine*, **4**, 18–23.

OZAKI, T., SUZUKI, T., FURUHASHI, T., OKUMA, S., and UCHIKAWA, Y., 1991, Trajectory control of robotic manipulators using neural networks. *IEEE Transactions on Industrial Electronics*, **38**, 195–202.

POGGIO, T., and GIROSI, F., 1990, Networks for approximation and learning. *Proceedings of IEEE*, **78**, 1481–1497.

RICE, J. R., 1964, *The Approximation of Functions* (Reading, Massachusetts: Addison-Wesley).

SANNER, R. M., and SLOTINE, J. E., 1992, Gaussian networks for direct adaptive control. *IEEE Transactions on Neural Networks*, **3**, 837–863.

SEPE, R. B., and LANG, J. H., 1991, Real-time adaptive control of the permanent magnet synchronous motor. *IEEE Transactions on Industry Applications*, **27**, 706–714.

SLOTINE, J. E., and LI, W., 1991, *Applied Nonlinear Control* (Englewood Cliffs, New Jersey: Prentice-Hall).

TZIRKEL, H. E., and FALLSIDE, F., 1991, A direct control method for a class of nonlinear systems using neural networks. Technical Report CUED/F-INFENG/TR65, University of Cambridge.

XU, J. X., Hashimoto, H., Slotine, J. J. E., Arai, Y., and HARASHIMA, F., 1989, Implementation of VSS control to robotic manipulators-smoothing modification. *IEEE Transactions on Industrial Electronics*, **36**, 321–329.

ZHANG, J., and BARTON, T. H., 1991, Robustness enhancement of dc drives with a smooth optimal sliding mode control. *IEEE Transactions on Industry Applications*, **27**, 686–693.