An Adaptive H^∞ Controller Design for Bank-To-Turn Missiles Using Ridge Gaussian Neural Networks

Chuan-Kai Lin and Sheng-De Wang, *Member, IEEE*

***Abstract*—A new autopilot design for bank-to-turn (BTT) missiles is presented. In the design of autopilot, a ridge Gaussian neural network with local learning capability and fewer tuning parameters than Gaussian neural networks is proposed to model the controlled nonlinear systems. We prove that the proposed ridge Gaussian neural network, which can be a universal approximator, equals the expansions of rotated and scaled Gaussian functions. Although ridge Gaussian neural networks can approximate the nonlinear and complex systems accurately, the small approximation errors may affect the tracking performance significantly. Therefore, by employing the H^∞ control theory, it is easy to attenuate the effects of the approximation errors of the ridge Gaussian neural networks to a prescribed level. Computer simulation results confirm the effectiveness of the proposed ridge Gaussian neural networks-based autopilot with H^∞ stabilization.**

***Index Terms*—Bank-to-turn (BTT) missiles, Gaussian neural networks, H^∞ control theory, ridge functions.**

NOMENCLATURE

A_x, A_y, A_z	Acceleration along the directions b_x, b_y, and b_z at center of mass, respectively.
$\{b_x, b_y, b_z\}$	A right-handed orthonormal basis of body coordinate frame, which is attached to the center of mass C of the missile, where b_x and b_y are on the longitudinal and lateral axis, respectively.
$C_{F_x}, C_{F_y}, C_{F_z}$	Aerodynamic force coefficients corresponding to the axes b_x, b_y, and b_z, respectively.
$C_{M_x}, C_{M_y}, C_{M_z}$	Moment coefficients corresponding to the axes b_x, b_y, and b_z, respectively.
d_p, d_q, d_r	Effective deflation angle, elevator deflation angle, and rudder deflation angle, respectively, (rad).
$\{e_x, e_y, e_z\}$	A right-handed orthonormal basis of inertial coordinate frame.
F_x, F_y, F_z	External forces along the axes b_x, b_y, and b_z, respectively, (N).
I_{xx}, I_{yy}, I_{zz}	Moment of inertia of the missile body about the axes b_x, b_y, and b_z, respectively, $(\mathrm{kg}\cdot\mathrm{m}^2)$.
l, m	Missile length (m) and missile mass (kg).
M_m	Mach number.
M_x, M_y, M_z	Total moment of inertia about the axes b_x, b_y, and b_z, respectively, $(\mathrm{N}\cdot\mathrm{m})$.
P, Q, R	Roll rate, pitch rate and yaw rate corresponding to the axes b_x, b_y, and b_z, respectively (clockwise), (rad/s).
Q_s	Dynamic pressure $(\mathrm{N}/\mathrm{m}^2)$.
S_{ref}	Aerodynamic reference area (m^2).
T	Thrust (kg).
V_s, V_m	Velocity of sound (m/s) and velocity of missile (m/s).
ω_a	Bandwidrh of actuator (rad/s).
$[U\ \ V\ \ W]^T$	Velocity vector of the missile transformed with respect to the body frame (m/s).
$[X\ \ Y\ \ Z]^T$	Position vector of the center of mass of the missile transformed with respect to the inertial frame (m).
α, β	Attack angle and sideslip angle, respectively, (rad).
δ_p, δ_q, δ_r	Actuator inputs for deflation angle, elevator deflation angle, and rudder deflation angle, respectively, (rad).
Φ	Body-axis roll angle measured from the downward vertical to b_z about the axis b_x (rad).
Θ	Body-axis pitch angle measured from the projection of b_x onto the horizontal plane to b_x (rad).
Ψ	Body-axis yaw angle measured between a fixed compass bearing and the projection of b_x onto the horizontal plane (rad).

I. INTRODUCTION

THE autopilot design for bank-to-turn (BTT) missiles has received considerable attention according to BTT missiles has higher maneuverability and aerodynamic acceleration compared with skid-to-turn missiles [10]. However, the requirement of high roll rate for BTT missiles to change the orientation of the acceleration will induce undesirable cross coupling between pitch and yaw motions [25]. Furthermore, the highly nonlinear aerodynamics and missile dynamics of nonminimum phase make the autopilot design more difficult.

A wide variety of approaches have been used successfully to address the autopilot design for missiles. Adaptive robust control based on well-known input/output (I/O) feedback linearization technique to achieve the satisfactory tracking performance have been presented in [10] and [11]. In [12], the gain-scheduling approach based on H^∞ control theory was proposed. In

Manuscript received June 2, 2003; revised July 23, 2003.

C.-K. Lin is with the Department of Electrical Engineering, Chinese Naval Academy, Kaohsiung 813, Taiwan (e-mail: cklin@mail.cna.edu.tw).

S.-D. Wang is with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: sdwang@hpc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TNN.2004.824418

the past decade, H^∞ optimal control has been widely discussed for robustness and its capability of disturbance rejection in linear and nonlinear control systems [6], [7], however, for partly unknown dynamics, the gain-scheduling for H^∞ autopilot was not satisfactory. Exploiting neural networks for BTT missiles control has been studied recently years [24]–[28]. In [25], although the hybrid radial basis function (RBF) network autopilot with localized learning capability has demonstrated better performance than gain scheduled autopilot, the adjustable parameters of RBFs are only the hidden-to-output weights.

As to H^∞ control theory combined with neural networks, not only optimal tracking can be achieved while perturbations are absent, but also the worst case effect on the tracking errors due to the parameter uncertainties and external disturbances can be reduced to be less than or equal a desired level [8], [9]. However, the input-to-hidden weights should be chosen by heuristic method due to only the hidden-to-output weights can be tuned in [8] and [9].

Motivated by the above approaches, we are inspired to integrate a proposed ridge Gaussian neural network, which is just a three-layer neural network with Gaussian activation functions, and H^∞ control theory to enhance the BTT missiles autopilot design in handling the tracking control problem with unmodeled uncertainties. It can be shown that the ridge Gaussian neural network is equivalent to the radial Gaussian neural network with matrices of scales and rotations of input vectors for each node. The advantages of ridge Gaussian neural network are fewer parameters to be tuned than traditional radial Gaussian neural network and both input-to-hidden and hidden-to-output weights can be on-line tuned.

The remainder of this paper is organized as follows. In Section II, the control objective is stated. Section III describes the ridge Gaussian neural networks. The H^∞ controller design using the ridge Gaussian neural networks is given in Section IV. In Section V, a highly nonlinear system, a BTT missile, is controlled by the H^∞ controller to demonstrate the availability of the proposed controller. At last, Section VI concludes the paper.

II. Problem Statement

The detailed dynamic equations and state notations of BTT missiles can be referred to the Appendix I. In general, the guidance of BTT missiles generates desired commands including desired rolling angle Φ_c, and desired accelerations A_{yc} and A_{zc}. It is obvious that the objective of the autopilot design is to drive the BTT missile to track the commands. However, the output assignment (Φ, A_y, A_z) will leave the BTT missile system with the undesirable nonminimum phase phenomenon, which will result in the I/O feedback linearization technique can not be applied directly to acceleration control of BTT missiles [12], [28]. The nonminimum phase characteristic can be circumvented by adopting redefined outputs as (Φ, α, β) or (Φ, V, W) [12] to apply I/O feedback linearization technique. According to $\alpha = \tan^{-1}(W/U)$ and $\beta = \tan^{-1}(V/U)$, there are several possible alternatives for the redefined output set (Φ, α, β) as (Φ, V, W), ($P\cos\alpha + R\sin\alpha$, α, β) and (μ, α, β), where μ is an aerodynamic bank angle [24]. In our approach, new output signals Φ, V and W are chosen according to the profile of the desired trajectories.

The nominal plant of BTT missiles can be rewritten by the input-output feedback linearization technique in the following form:

$$\begin{bmatrix} y_1^{(n_1)} \\ y_2^{(n_2)} \\ y_3^{(n_3)} \end{bmatrix} = \begin{bmatrix} a_1(\boldsymbol{x}) \\ a_2(\boldsymbol{x}) \\ a_3(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad or$$

$$\boldsymbol{y}^{(n)} = \boldsymbol{a}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} \tag{1}$$

where $\boldsymbol{y} = [y_1 \;\; y_2 \;\; y_3]^T = [\Phi \;\; V \;\; W]^T$, $[u_1 \;\; u_2 \;\; u_3]^T = [\delta_p \;\; \delta_q \;\; \delta_r]^T$ and $\boldsymbol{x} = [P \;\; Q \;\; R \;\; \Phi \;\; \Theta \;\; \Psi \;\; U \;\; V \;\; W]^T$. The relative degree (n_1, n_2, n_3) for the three input-output channels is found to be (3, 2, 2). In the above equation, $\boldsymbol{a}(\boldsymbol{x})$ is an unknown function vector, $\boldsymbol{B}(\boldsymbol{x})$ is a known gain matrix as given in the Appendix I and $\boldsymbol{u}$ is the input vector. It is also assumed that zero dynamics are exponentially stable.

For alleviating the nonminimum phase phenomenon, output-redefinition method is applied such that the new command set (Φ_c, V_c, W_c) should be transformed from the original command set (Φ_c, A_{yc}, A_{zc}). After transforming commands, Φ_c is not changed, $V_c = 0$ and $W_c = K_w(A_z - A_{zc})$ with a constant K_w. In practice, the desired trajectories Φ_c (for Φ) and W_d (for W), which should be smooth, are generated by two first-order filters with inputs Φ_c and W_c. The desired trajectory for V, V_d, still remains zero for keeping small sideslip angle β due to $\beta = \tan^{-1}(V/U)$.

Then, the tracking errors are defined as follows:

$$e_1 = y_1 - y_{1d} = \Phi - \Phi_d \tag{2a}$$

$$e_2 = y_2 - y_{2d} = V - V_d \tag{2b}$$

$$e_3 = y_3 - y_{3d} = W - W_d \tag{2c}$$

where y_{1d}, y_{2d}, and y_{3d} are the desired trajectories of y_1, y_2, and y_3, respectively. Hence, a sliding-surface vector $\boldsymbol{s}(t)$ is defined as

$$\begin{aligned} \boldsymbol{s}(t) &= [s_1(t) \quad s_2(t) \quad s_3(t)]^T \\ &= \left[\ddot{e}_1 + 2k_1\dot{e}_1 + k_1^2 e_1 \quad \dot{e}_2 + k_2 e_2 \quad \dot{e}_3 + k_3 e_3\right]^T \end{aligned} \tag{3}$$

where each k_j is a strictly positive constant for $1 \le j \le 3$. Differentiating $\boldsymbol{s}(t)$ with respect to time, we can get

$$\dot{\boldsymbol{s}}(t) = \begin{bmatrix} y_1^{(3)} \\ y_2^{(2)} \\ y_3^{(2)} \end{bmatrix} + \begin{bmatrix} -y_{1d}^{(3)} + 2k_1\ddot{e}_1 + k_1^2\dot{e}_1 \\ -y_{2d}^{(2)} + k_2\dot{e}_2 \\ -y_{3d}^{(2)} + k_3\dot{e}_3 \end{bmatrix}$$

or

$$\dot{\boldsymbol{s}}(t) = \boldsymbol{y}^{(n)} + \boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{a}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{r}(\boldsymbol{x}) \tag{4}$$

where $\boldsymbol{r}(\boldsymbol{x}) = [-y_{1d}^{(3)} + 2k_1\ddot{e}_1 + k_1^2\dot{e}_1 \;\; - y_{2d}^{(2)} + k_2\dot{e}_2 \;\; - y_{3d}^{(2)} + k_3\dot{e}_3]^T$.

Assume the inverse of known matrix $\boldsymbol{B}(\boldsymbol{x})$, $\boldsymbol{B}^{-1}(\boldsymbol{x})$, exists and the unknown $\boldsymbol{a}(\boldsymbol{x})$ is approximated by $\hat{\boldsymbol{a}}(\boldsymbol{x})$, then the control law can be of the following form:

$$\boldsymbol{u} = \boldsymbol{B}^{-1}(\boldsymbol{x})\left[-\hat{\boldsymbol{a}}(\boldsymbol{x}) - \boldsymbol{r}(\boldsymbol{x}) - \boldsymbol{K}_s\boldsymbol{s}(t) - \hat{\boldsymbol{d}}\right] \tag{5}$$

where $\boldsymbol{K}_s = \text{diag}\{K_{s1}, K_{s2}, K_{s3}\} > 0$ and $\hat{\boldsymbol{d}}$ is used to attenuate approximation errors. Substituting (5) into (4) yields

$$\dot{\boldsymbol{s}}(t) = -\boldsymbol{K}_s \boldsymbol{s}(t) + \boldsymbol{d}_a - \hat{\boldsymbol{d}} \tag{6}$$

where $\boldsymbol{d}_a = \boldsymbol{a}(\boldsymbol{x}) - \hat{\boldsymbol{a}}(\boldsymbol{x})$. The estimation $\hat{\boldsymbol{a}}(\boldsymbol{x})$, which can approximate $\boldsymbol{a}(\boldsymbol{x})$ accurately, is implemented by the proposed ridge Gaussian neural network described in the next section. Due to the appearance of approximation error $\boldsymbol{d}_a$, our control goal is to design an H^∞ BTT autopilot with which the BTT missile system can suppress the approximation error and follow the desired commands while maintaining the small sideslip angle. The H^∞ tracking performance represented in terms of a finite L_2 gain relationship is expressed as follows [8]:

$$\int_0^T \|\boldsymbol{s}\|_Q^2 dt \le \gamma^2 \int_0^T \|\boldsymbol{d}_a\|^2 dt + \beta_H \tag{7}$$

where β_H, γ, $T \in \Re^+$ and $\boldsymbol{d}_a \in L_{2e}$, i.e., $\int_0^T \|\boldsymbol{d}_a\|^2 dt \le \infty$ (L_{2e} is the extended L_2 space). Therefore, our control objective is to design a ridge Gaussian neural network-based autopilot for BTT missiles to achieve the above H^∞ tracking performance.

III. Ridge Gaussian Neural Networks

A basic building block of nearly all artificial neural networks is an adaptive linear combiner which computes the sum of the weighted inputs [13]. The Gaussian neural networks as other radial neural networks lack of the adaptive linear combiner, however, the general multilayed neural networks with sigmoid functions lack of the local learning capability. Therefore, another purpose of the ridge Gaussian neural networks is to show that ridge Gaussian neural networks can be also represented by an architecture consisting of adaptive linear combiners cascaded with Gaussian activation functions and take advantage of local learning capability.

A real-valued function f defined on $\Re^l$ is said to be a ridge function if there exist a linear functional $\boldsymbol{v}$: $\Re^l \to \Re$ and a function ϕ: $\Re \to \Re$ such that $f = \phi \circ \boldsymbol{v}$ [14]. Thus, f has the form $f = \phi(\langle \boldsymbol{x}, \boldsymbol{v} \rangle) = \phi(v_0 x_0 + v_1 x_1 + \cdots + v_n x_n)$. Therefore, the three-layered neural networks can be viewed as superpositions of ridge functions. With the ridge functions chosen as sigmoid activations, it is just the feedforward multi-layered neural network, which has been proved to be an accurate approximator in [15] and [16]. Different ridge functions as Gaussian function should be considered, however, the proof in [15] and [16] restricted the set of monotone increasing ridge functions.

The idea using Gaussian functions as activation functions of neural networks is good but needs theoretical support. Now, let us try to formulate what we have in mind more clearly. Given an input vector $\boldsymbol{x}_{NN} = [1 \;\; \boldsymbol{x}]^T$ in $\Re^{l+1}$, the output of a ridge Gaussian neural network is determined by

$$a_i = \sum_{j=1}^{N_h} w_{ij} \phi \left(\sum_{k=1}^{l} v_{jk} x_k + v_{j0} \right), \quad i = 1, \ldots, q \tag{8}$$

with $\phi(\cdot) = m(t) e^{-(\cdot)^2}$ the Gaussian ridge function, w_{ij} the hidden-to-output layer weights, v_{jk} the input-to-hidden layer weights, and N_h the hidden layer neuron number. According to the amplitude modification $m(t) = e^{-\alpha_j \boldsymbol{x}_{NN}^T \boldsymbol{x}_{NN}}$ should be positive and not greater than 1, α_js should be positive.

Rewrite (3.1) the matrix expression as

$$\boldsymbol{a} = \boldsymbol{W}^T \boldsymbol{\phi}(\boldsymbol{V}^T \boldsymbol{x}_{NN}) \tag{9}$$

where hidden-to-output weight matrix is $\boldsymbol{W} = [w_{ij}]$, and input-to-hidden weight matrix is $\boldsymbol{V} = [v_{jk}]$. At first, the definition of traditional radial Gaussian basis functions is given.

Definition 1: Define radial Gaussian basis functions as

$$\phi_j(\|\boldsymbol{x} - \boldsymbol{c}_j\|, \boldsymbol{\sigma}_j) = e^{-(\boldsymbol{x}-\boldsymbol{c}_j)^T \boldsymbol{S}_j (\boldsymbol{x}-\boldsymbol{c}_j)}, \quad j = 1, \ldots, N_h \tag{10}$$

where $\boldsymbol{x} = [x_1 \;\; \cdots \;\; x_l]^T$, $\boldsymbol{c}_j = [c_{j1} \;\; \cdots \;\; c_{jl}]^T$ is the center vector of radial Gaussian basis functions and $\boldsymbol{S}_j = \text{diag}((1/\sigma_{j1}^2), \cdots, (1/\sigma_{jl}^2))$.

The following theorem proves that the ridge Gaussian neural network is equivalent to the sum of scaled and rotated radial Gaussian basis functions.

Theorem 1: A three-layer ridge Gaussian neural network is equalivent to a three-layer neural network with rotated Gaussian activation functions.

Proof: In (8), let $m(t) = e^{-\alpha_j \boldsymbol{x}_{NN}^T \boldsymbol{x}_{NN}}$ such that each output of ridge Gaussian activation functions can be written as

$$\begin{aligned} \phi_j &= e^{-\alpha_j \boldsymbol{x}_{NN}^T \boldsymbol{x}_{NN}} e^{-(v_{j0} + v_{j1} x_1 + \cdots + v_{jn} x_n)^2} \\ &= e^{-\boldsymbol{x}_{NN}^T (\alpha_j \boldsymbol{I} + \boldsymbol{v}_j \boldsymbol{v}_j^T) \boldsymbol{x}_{NN}}, \quad j = 1, \ldots, N_h \end{aligned} \tag{11}$$

where ϕ_j is the jth element of $\boldsymbol{\phi}$ and $\boldsymbol{V}_j$ is the jth column of $\boldsymbol{V}$. If we can prove that ridge Gaussian basis functions can be represented as

$$\begin{aligned} &\phi_j\left(\|\boldsymbol{x} - \boldsymbol{\eta}_j\|, \boldsymbol{\sigma}_j\right) \\ &\quad = m(t) e^{-(\boldsymbol{x}-\boldsymbol{\eta}_j)^T \boldsymbol{R}_j^T \boldsymbol{S}_j \boldsymbol{R}_j (\boldsymbol{x}-\boldsymbol{\eta}_j)} \\ &\quad = m(t) e^{-(\boldsymbol{z}-\boldsymbol{\varsigma}_j)^T \boldsymbol{S}_j (\boldsymbol{z}-\boldsymbol{\varsigma}_j)}, \quad j = 1, \ldots, N_h \end{aligned} \tag{12}$$

where $m(t)$ is the amplitude modification, $\boldsymbol{R}_j$ is the rotation matrix, $\boldsymbol{z} = \boldsymbol{R}_j \boldsymbol{x}$ is the rotated input vector and $\boldsymbol{\varsigma}_j = \boldsymbol{R}_j \boldsymbol{\eta}_j$ is the center vector, then the proof of *Theorem 1* is completed. According to $\alpha_j \boldsymbol{I} + \boldsymbol{V}_j \boldsymbol{V}_j^T$ is a symmetric matrix, there exists a matrix $\boldsymbol{\Gamma}_j$ with orthogonal columns such that

$$\alpha_j \boldsymbol{I} + \boldsymbol{V}_j \boldsymbol{V}_j^T = \boldsymbol{\Gamma}_j^T \bar{\boldsymbol{\Lambda}}_j \boldsymbol{\Gamma}_j, \quad j = 1, \ldots, N_h \tag{13}$$

where $\bar{\boldsymbol{\Lambda}}_j = \text{diag}(\lambda_0, \lambda_1, \cdots, \lambda_l)$. The matrix $\boldsymbol{\Gamma}_j$ can be divided into four blocks

$$\boldsymbol{\Gamma}_j^T = \begin{bmatrix} \boldsymbol{\Gamma}_{11} & \boldsymbol{\Gamma}_{12} \\ \boldsymbol{\Gamma}_{21} & \boldsymbol{\Gamma}_{22} \end{bmatrix} \tag{14}$$

where $\boldsymbol{\Gamma}_{11}$ is 1 by 1, $\boldsymbol{\Gamma}_{12}$ is 1 by n, $\boldsymbol{\Gamma}_{21}$ is n by 1 and $\boldsymbol{\Gamma}_{22}$ is n by n matrix. Thus (11) can be rewritten as

$$\begin{aligned} \phi_j = e^{-\lambda_0 (\boldsymbol{\Gamma}_{11} + \boldsymbol{\Gamma}_{12} \boldsymbol{x})^2} e^{-(\boldsymbol{\Gamma}_{21} + \boldsymbol{\Gamma}_{22} \boldsymbol{x})^T \boldsymbol{\Lambda}_j (\boldsymbol{\Gamma}_{21} + \boldsymbol{\Gamma}_{22} \boldsymbol{x})}, \\ j = 1, \ldots, N_h \end{aligned} \tag{15}$$

where $m(t) = e^{-\lambda_0 (\boldsymbol{\Gamma}_{11} + \boldsymbol{\Gamma}_{12} \boldsymbol{x})^2}$, $\boldsymbol{\Lambda}_j = \text{diag}(\lambda_1, \cdots, \lambda_l)$, the rotated input vector $\boldsymbol{z} = \boldsymbol{\Gamma}_{22} \boldsymbol{x}$, the center vector $\boldsymbol{\varsigma}_j = -\boldsymbol{\Gamma}_{21}$ and the rotation matrix $\boldsymbol{R}_j = \boldsymbol{\Gamma}_{22}$. ■

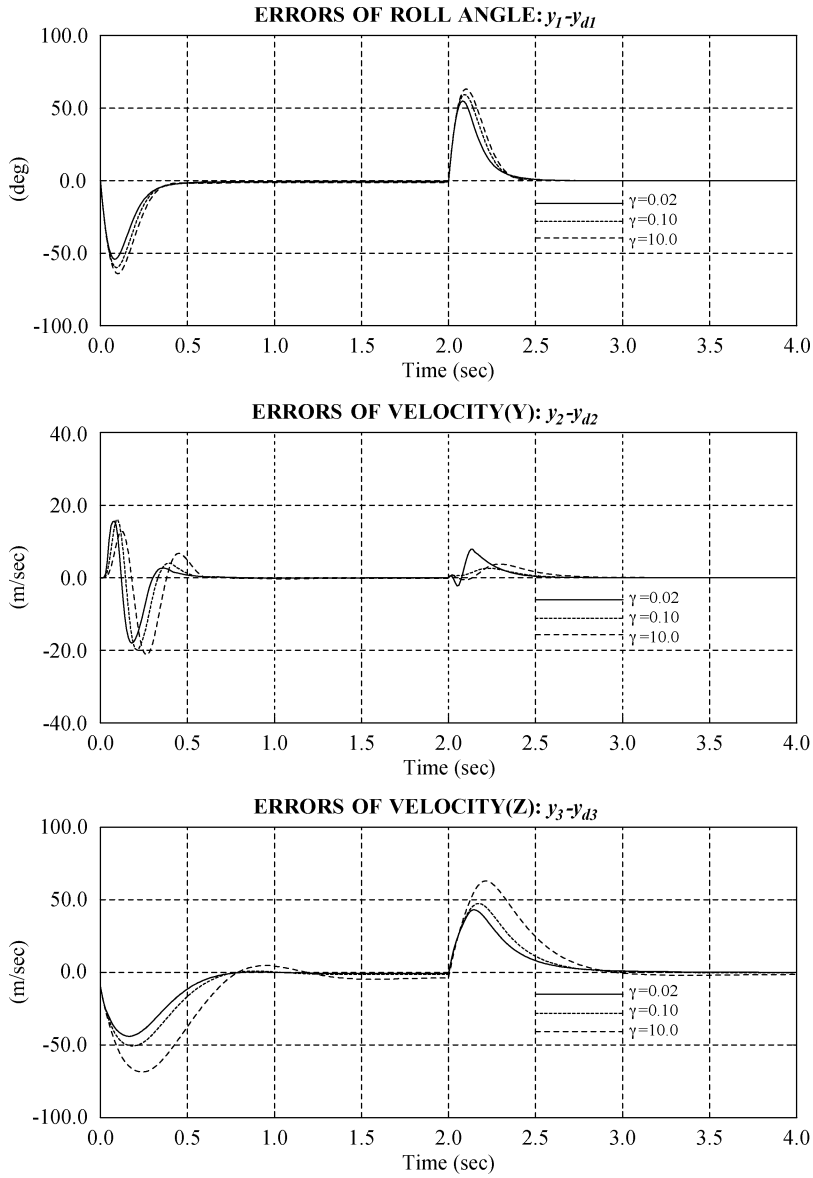

Fig. 1. The tracking errors of redefined outputs with different γ.

Besides from the local learning capability, the ridge Gaussian neural network has other advantages. To avoid tuning α_js to be negative, α_js are selected as small positive values. Thus, a traditional Gaussian neural network [17] with p input variables has N_h centers, N_h variances and $N_h \times q$ hidden-to-output layer interconnection weights, that is, $2 \times p \times N_h + N_h \times q$ parameters. As to the ridge Gaussian neural network, the total number of parameters is $p \times N_h + N_h \times q$. In other words, the ridge Gaussian neural network has fewer parameters to be tuned. Another drawback of Gaussian neural networks is that guaranteeing the capability of reconstruction makes the size of the network sparse and large.

In consequence of using ridge Gaussian neural networks as approximators, we will show that the ridge Gaussian neural networks can approximate any nonlinear function to any desired accuracy.

Theorem 2: Let $\phi_j(\boldsymbol{x}) = e^{-\alpha_j \boldsymbol{x}_{NN}^T \boldsymbol{x}_{NN}} e^{-\boldsymbol{x}_{NN}^T \boldsymbol{V}_j \boldsymbol{V}_j^T \boldsymbol{x}_{NN}}$ and S' be a compact metric space, $C(S')$ be the set of all continuous real-valued functions on S', and A_s be a subset of $C(S')$. If

1) A_s is an algebra;
2) A_s separates points of S';
3) A_s contains the constant functions, then $\bar{A}_s$ the closure of A_s, That is, given $\varepsilon > 0$ and a function g on S', we can find a function in (8) $a_i \in A_s$ such that $\max_{\boldsymbol{x} \in A_s} ||a_i(\boldsymbol{x}) - g(\boldsymbol{x})|| < \varepsilon$.

The proof of this theorem given in Appendix II, which is similar to [3] and based on Stone-Weierstrass theorem, is just to examine that the set $A_s = \{a_i | a_i = \sum_{j=1}^{N_h} w_{ij} \phi(\sum_{k=1}^{l} v_{jk} x_k + v_{j0})\}$ satisfies the three conditions of *Theorem* 2 (or three conditions of the Stone–Weierstrass theorem [18]). In other words, this theorem states that ridge Gaussian neural networks are universal approximators.

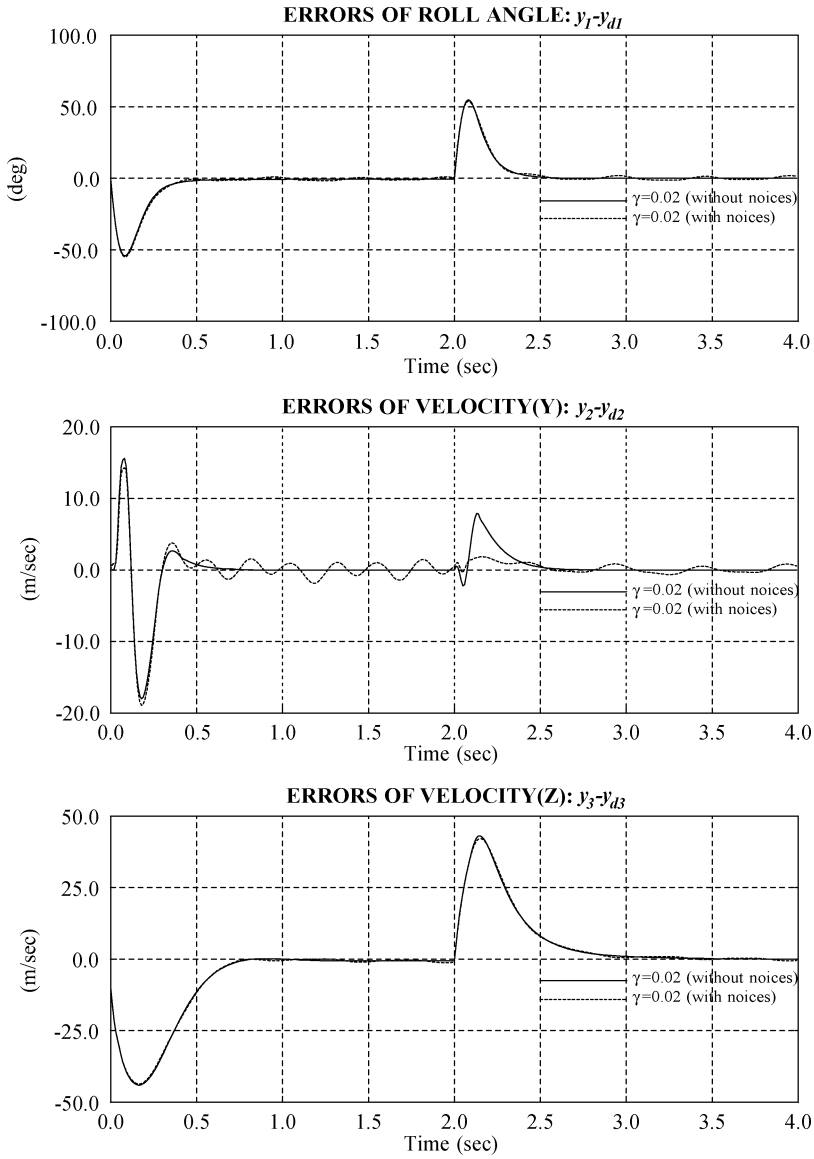

Fig. 2. Comparisons of the tracking errors of redefined outputs with and without noises.

IV. Controller Design

In the controller design, a ridge Gaussian neural network is employed to approximate $a(\boldsymbol{x})$. From *Theorem* 2, the ridge Gaussian neural networks with sufficient number of nodes can approximate any function with arbitrary accuracy. Therefore, an assumption is given as follows.

Assumption 1: There exist ideal matrices $\bar{\boldsymbol{V}}$ and $\bar{\boldsymbol{W}}$ such that $\boldsymbol{a}(\boldsymbol{x}|\bar{\boldsymbol{V}},\bar{\boldsymbol{W}}) = \bar{\boldsymbol{W}}^T\boldsymbol{\phi}(\bar{\boldsymbol{V}}^T\boldsymbol{x}_{NN})$ with $\boldsymbol{x}_{NN} = [1 \;\; \boldsymbol{x}]^T$ can approximate $\boldsymbol{a}(\boldsymbol{x})$ with accuracy ε over a compact set S', that is, $\exists \bar{\boldsymbol{V}}, \bar{\boldsymbol{W}}$ such that $\sup_{\boldsymbol{x}\in A_s} \|\boldsymbol{a}(\boldsymbol{x}|\bar{\boldsymbol{V}},\bar{\boldsymbol{W}}) - \boldsymbol{a}(\boldsymbol{x})\| \le \varepsilon$.

For ease of notation, we denote $\boldsymbol{\phi} = \boldsymbol{\phi}(\boldsymbol{V}^T\boldsymbol{x}_{NN})$ and $\hat{\boldsymbol{\phi}} = \hat{\boldsymbol{\phi}}(\boldsymbol{V}^T\boldsymbol{x}_{NN})$. By employing weight updating law, $\hat{\boldsymbol{V}}$ and $\hat{\boldsymbol{W}}$ can approach the optimal weight matrices $\boldsymbol{V}$ and $\boldsymbol{W}$. Applying $\boldsymbol{a}(\boldsymbol{x})-\hat{\boldsymbol{a}}(\boldsymbol{x}) = \boldsymbol{a}(\boldsymbol{x})-\boldsymbol{a}(\boldsymbol{x}|\bar{\boldsymbol{V}},\bar{\boldsymbol{W}})+\boldsymbol{a}(\boldsymbol{x}|\bar{\boldsymbol{V}},\bar{\boldsymbol{W}})-\hat{\boldsymbol{a}}(\boldsymbol{x})= \boldsymbol{\varepsilon}(\boldsymbol{x})+ \boldsymbol{W}^T\boldsymbol{\phi}(\boldsymbol{V}^T\boldsymbol{x}_{NN}) - \hat{\boldsymbol{W}}^T\boldsymbol{\phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}_{NN})= \boldsymbol{W}^T\tilde{\boldsymbol{\phi}} + \tilde{\boldsymbol{W}}^T\hat{\boldsymbol{\phi}} + \boldsymbol{\varepsilon}(\boldsymbol{x})$ to (6), we obtain

$$\dot{\boldsymbol{s}}(t) = -\boldsymbol{K}_s\boldsymbol{s}(t) + (\boldsymbol{W}^T\tilde{\boldsymbol{\phi}} + \tilde{\boldsymbol{W}}^T\hat{\boldsymbol{\phi}}) + \boldsymbol{\varepsilon}(\boldsymbol{x}) - \hat{\boldsymbol{d}} \tag{16}$$

where $\tilde{\boldsymbol{W}} = \boldsymbol{W} - \hat{\boldsymbol{W}}$, $\tilde{\boldsymbol{\phi}} = \boldsymbol{\phi}(\boldsymbol{V}^T\boldsymbol{x}_{NN}) - \boldsymbol{\phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}_{NN})$ and $\hat{\boldsymbol{\phi}} = \boldsymbol{\phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}_{NN})$. By Taylor series expansion [19], rewrite $\tilde{\boldsymbol{\phi}}$ as

$$\tilde{\boldsymbol{\phi}}=\boldsymbol{\phi}(\boldsymbol{V}^T\boldsymbol{x}_{NN})-\boldsymbol{\phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}_{NN})=\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{V}}^T\boldsymbol{x}_{NN}+O\left((\tilde{\boldsymbol{V}}^T\boldsymbol{x}_{NN})^2\right) \tag{17}$$

where $\tilde{\boldsymbol{V}} = \boldsymbol{V} - \hat{\boldsymbol{V}}$, $\hat{\boldsymbol{\phi}}' = \operatorname{diag}\{\hat{\phi}'_1, \cdots, \hat{\phi}'_{N_h}\}$ and $O((\tilde{\boldsymbol{V}}^T\boldsymbol{x}_{NN})^2)$ denotes the Taylor series higher-order terms. Substituting (17) in (16) yields

$$\dot{\boldsymbol{s}}(t)=-\boldsymbol{K}_s\boldsymbol{s}(t)+\tilde{\boldsymbol{W}}^T(\hat{\boldsymbol{\phi}}-\hat{\boldsymbol{\phi}}'\hat{\boldsymbol{V}}^T\boldsymbol{x}_{NN})+\hat{\boldsymbol{W}}^T\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{V}}^T\boldsymbol{x}_{NN}+\boldsymbol{d}-\hat{\boldsymbol{d}} \tag{18}$$

where $\boldsymbol{d} = \tilde{\boldsymbol{W}}^T\hat{\boldsymbol{\phi}}'\boldsymbol{V}^T\boldsymbol{x}_{NN} + \boldsymbol{W}^TO((\tilde{\boldsymbol{V}}^T\boldsymbol{x}_{NN})^2) + \boldsymbol{\varepsilon}(\boldsymbol{x})$. Take initial conditions $\boldsymbol{s}(0)$, $\tilde{\boldsymbol{V}}(0)$ and $\tilde{\boldsymbol{W}}(0)$ into consider, the H^∞ tracking performance [9] is modified as

$$\int_0^T \boldsymbol{s}^T\boldsymbol{Q}\boldsymbol{s}dt \le \beta + \gamma^2\int_0^T \boldsymbol{d}^T\boldsymbol{d}dt, \quad \forall T\in[0,\infty)\; \boldsymbol{d}\in L_2[0,T] \tag{19}$$

where $\beta = \boldsymbol{s}^T(0)\boldsymbol{P}\boldsymbol{s}(0) + \mathrm{tr}\{\tilde{\boldsymbol{V}}^T(0)\boldsymbol{K}_V^{-1}\tilde{\boldsymbol{V}}^T(0)\} + \mathrm{tr}\{\tilde{\boldsymbol{W}}^T(0)\boldsymbol{K}_W^{-1}\tilde{\boldsymbol{W}}^T(0)\}$, $\boldsymbol{Q} = \boldsymbol{Q}^T \geq 0$, $\boldsymbol{P} = \boldsymbol{P}^T \geq 0$ and $\boldsymbol{K}_V^{-1}$ and $\boldsymbol{K}_W^{-1}$ are diagonal matrices. Then the control objective is to determine $\hat{\boldsymbol{d}}$, and the weight update law of $\hat{\boldsymbol{V}}$ and $\hat{\boldsymbol{W}}$ to guarantee the tracking performance (18) with a prescribed disturbance attenuation level γ.

Theorem 3: Applying the control law (5) to the nonlinear system (1) with

$$\dot{\hat{\boldsymbol{V}}} = \boldsymbol{K}_V \boldsymbol{x}_{NN} \boldsymbol{s}^T \boldsymbol{P} \hat{\boldsymbol{W}}^T \hat{\boldsymbol{\phi}}' \tag{20}$$

$$\dot{\hat{\boldsymbol{W}}} = \boldsymbol{K}_W (\hat{\boldsymbol{\phi}} - \hat{\boldsymbol{\phi}}' \hat{\boldsymbol{V}} \boldsymbol{x}_{NN}) \boldsymbol{s}^T \boldsymbol{P} \tag{21}$$

$$\hat{\boldsymbol{d}} = \frac{1}{k_d} \boldsymbol{P}\boldsymbol{s} \tag{22}$$

where $\boldsymbol{K}_V$ and $\boldsymbol{K}_W$ are diagonal matrices, and k_d is a positive constant. The matrix $\boldsymbol{P}$ is the solution of

$$Q = \boldsymbol{K}_s^T \boldsymbol{P} + \boldsymbol{P}\boldsymbol{K}_s + \left(\frac{2}{k_d} - \frac{1}{\gamma^2}\right)\boldsymbol{P}\boldsymbol{P} \tag{23}$$

Then the H^∞ tracking performance is achieved.

Proof: Select a storage function candidate

$$L = \frac{1}{2}\boldsymbol{s}^T \boldsymbol{P}\boldsymbol{s} + \frac{1}{2}\mathrm{tr}\left\{\tilde{\boldsymbol{V}}^T \boldsymbol{K}_V^{-1} \tilde{\boldsymbol{V}}\right\} + \frac{1}{2}\mathrm{tr}\left\{\tilde{\boldsymbol{W}}^T \boldsymbol{K}_W^{-1} \tilde{\boldsymbol{W}}\right\}. \tag{24}$$

The time derivative of L can be written as

$$\dot{L} = \frac{1}{2}\dot{\boldsymbol{s}}^T \boldsymbol{P}\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T \boldsymbol{P}\dot{\boldsymbol{s}} - \mathrm{tr}\left\{\tilde{\boldsymbol{V}}^T \boldsymbol{K}_V^{-1} \dot{\hat{\boldsymbol{V}}}\right\} - \mathrm{tr}\left\{\tilde{\boldsymbol{W}}^T \boldsymbol{K}_W^{-1} \dot{\hat{\boldsymbol{W}}}\right\}. \tag{25}$$

Substituting (18) and (20)–(22) in (25) yields

$$\begin{aligned}
\dot{L} = \frac{1}{2}\Big[&-\boldsymbol{s}^T \boldsymbol{K}_s^T + (\hat{\boldsymbol{\phi}} - \hat{\boldsymbol{\phi}}' \hat{\boldsymbol{V}}^T \boldsymbol{x}_{NN})^T \tilde{\boldsymbol{W}} \\
&+ \boldsymbol{x}_{NN}^T \tilde{\boldsymbol{V}} \hat{\boldsymbol{\phi}}'^T \hat{\boldsymbol{W}} + \boldsymbol{d}^T - \hat{\boldsymbol{d}}^T\Big] \boldsymbol{P}\boldsymbol{s} \\
&+ \frac{1}{2}\boldsymbol{s}^T \boldsymbol{P}\Big[-\boldsymbol{K}_s \boldsymbol{s} + \tilde{\boldsymbol{W}}^T (\hat{\boldsymbol{\phi}} - \hat{\boldsymbol{\phi}}' \hat{\boldsymbol{V}}^T \boldsymbol{x}_{NN}) \\
&\quad + \hat{\boldsymbol{W}}^T \hat{\boldsymbol{\phi}}' \tilde{\boldsymbol{V}}^T \boldsymbol{x}_{NN} + \boldsymbol{d} - \hat{\boldsymbol{d}}\Big] \\
&- \mathrm{tr}\left\{(\tilde{\boldsymbol{V}}^T \boldsymbol{x}_{NN}) \boldsymbol{s}^T \boldsymbol{P} (\hat{\boldsymbol{W}}^T \hat{\boldsymbol{\phi}}')\right\} \\
&- \mathrm{tr}\left\{\tilde{\boldsymbol{W}}^T (\hat{\boldsymbol{\phi}} - \hat{\boldsymbol{\phi}}' \hat{\boldsymbol{V}}^T \boldsymbol{x}_{NN}) \boldsymbol{s}^T \boldsymbol{P}\right\}.
\end{aligned} \tag{26}$$

Since $\mathrm{tr}\{\tilde{\boldsymbol{W}}^T (\hat{\boldsymbol{\phi}} - \hat{\boldsymbol{\phi}}' \hat{\boldsymbol{V}}^T \boldsymbol{x}_{NN}) \boldsymbol{s}^T \boldsymbol{P}\} = \boldsymbol{s}^T \boldsymbol{P} \tilde{\boldsymbol{W}}^T (\hat{\boldsymbol{\phi}} - \hat{\boldsymbol{\phi}}' \hat{\boldsymbol{V}}^T \boldsymbol{x}_{NN})$ and $\mathrm{tr}\{(\tilde{\boldsymbol{V}}^T \boldsymbol{x}_{NN}) \boldsymbol{s}^T \boldsymbol{P} (\hat{\boldsymbol{W}}^T \hat{\boldsymbol{\phi}}')\} = \boldsymbol{s}^T \boldsymbol{P} \hat{\boldsymbol{W}}^T \hat{\boldsymbol{\phi}}' (\tilde{\boldsymbol{V}}^T \boldsymbol{x}_{NN})$, (26) becomes

$$\begin{aligned}
\dot{L} &= \frac{1}{2}\boldsymbol{s}^T \left[-\boldsymbol{K}_s^T \boldsymbol{P} - \boldsymbol{P}\boldsymbol{K}_s - \left(\frac{2}{k_d} - \frac{1}{\gamma^2}\right)\boldsymbol{P}\boldsymbol{P}\right] \boldsymbol{s} \\
&\quad - \frac{1}{2}\left(\frac{1}{\gamma}\boldsymbol{P}\boldsymbol{s} - \gamma \boldsymbol{d}\right)^T \left(\frac{1}{\gamma}\boldsymbol{P}\boldsymbol{s} - \gamma \boldsymbol{d}\right) + \frac{1}{2}\gamma^2 \boldsymbol{d}^T \boldsymbol{d} \\
&\leq -\frac{1}{2}\boldsymbol{s}^T \boldsymbol{Q}\boldsymbol{s} + \frac{1}{2}\gamma^2 \boldsymbol{d}^T \boldsymbol{d}.
\end{aligned} \tag{27}$$

To complete the proof, it is necessary to integrate the both side of the above inequality from $t = 0$ to $t = T$

$$L(T) - L(0) \leq -\frac{1}{2}\int_0^T \boldsymbol{s}^T \boldsymbol{Q}\boldsymbol{s} dt + \frac{1}{2}\int_0^T \gamma^2 \boldsymbol{d}^T \boldsymbol{d} dt. \tag{28}$$

TABLE I
COMMAND SET VERSUS TIME

Time t (sec)	$\Phi_c(deg)$	$A_{yc}(g)$	$A_{zc}(g)$
$0 \leq t < 1.2$	0	0	-50
$1.2 \leq t < 3.0$	0	0	0
$3.0 \leq t < 4.8$	135	0	-20
$4.8 \leq t < 9.0$	0	0	0
$9.0 \leq t < 11.0$	90	0	-10
$11.0 \leq t < 13.0$	0	0	0
$13.0 \leq t < 16.0$	0	0	-5
$16.0 \leq t \leq 18.0$	0	0	0

Since $L(T) \geq 0$, (28) becomes

$$\begin{aligned}
\int_0^T \boldsymbol{s}^T \boldsymbol{Q}\boldsymbol{s} dt \leq{}& \boldsymbol{s}^T(0)\boldsymbol{P}\boldsymbol{s}(0) + \mathrm{tr}\left\{\tilde{\boldsymbol{V}}^T(0)\boldsymbol{K}_V^{-1}\tilde{\boldsymbol{V}}(0)\right\} \\
&+ \mathrm{tr}\left\{\tilde{\boldsymbol{W}}^T(0) K_W^{-1} \tilde{\boldsymbol{W}}(0)\right\} + \gamma^2 \int_0^T \boldsymbol{d}^T \boldsymbol{d} dt
\end{aligned} \tag{29}$$

that is, (19) is obtained. ■

Remark: Since $\boldsymbol{K}_s$ is a diagonal matrix, the Riccati-like (23) can be solved simply. Let $\boldsymbol{P}$ and $\boldsymbol{Q}$ be diagonal matrices, and $(2/k_d) - (1/\gamma^2) \geq 0$, then (23) is simplified to three simple quadratic equations. Therefore, (23) can be solved easily by choosing k_d, γ, and $\boldsymbol{Q}$ carefully.

V. SIMULATION RESULTS

In the gain scheduling controller [12], the controller is highly dependent on the LTI controller at each fixed operating point. It is of practical significance for a gain-scheduling controller with a fewer number of linear controllers [28]. In our approach, complexity is reduced for only one autopilot is used. Therefore, our approach is simpler and easier to design.

In the following simulations, the BTT missile is subject to the following physical limitations:

1) attack angle α: $\alpha \geq 0°$;
2) sideslip angle β: $|\beta| \leq 5°$;
3) roll rate P: $|P| \leq 500°/\mathrm{s}$;
4) pitch rate Q: $|Q| \leq 150°/\mathrm{s}$;
5) yaw rate R: $|R| \leq 150°/\mathrm{s}$;
6) actuator position saturation: $|d_p| \leq 20°$, $|d_q| \leq 20°$ and $|d_r| \leq 20°$.

In the first simulation, during the time interval between 0 and 2 s, the desired output signals were $\Phi_c = 135°$, $A_{yc} = 0$ g and $A_{zc} = -15$ g ($g = 9.8\ \mathrm{m/s^2}$). And $\Phi_c = A_{yc} = A_{zc} = 0$ for 2∼4 s. For obtaining desired smooth trajectories, Φ_d and W_d were generated by $\dot{\Phi}_d = -10\Phi_d + 10\Phi_c$ and $\dot{W}_d = -10W_d + 10(A_z - A_{zc})$. The number of neurons in the hidden layer was chosen as 20. If we use the method in [2], then the size will be over 2000 neurons. The initial conditions are $\mathrm{height} = -Z = 5000\ \mathrm{m}$ and $\boldsymbol{x}(0) = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1020\ (\mathrm{m/s})\ \ 0\ \ -0.01\ (\mathrm{m/s})]^T$. In addition, the design parameters were selected as $\mathrm{Q} = \mathrm{diag}(10, 10, 10)$ and three different $(\gamma,\ k_d)$ pairs: $(\gamma, k_d) = (0.02, 0.0005)$, (0.1, 0.01), and (10.0, 2.0). The

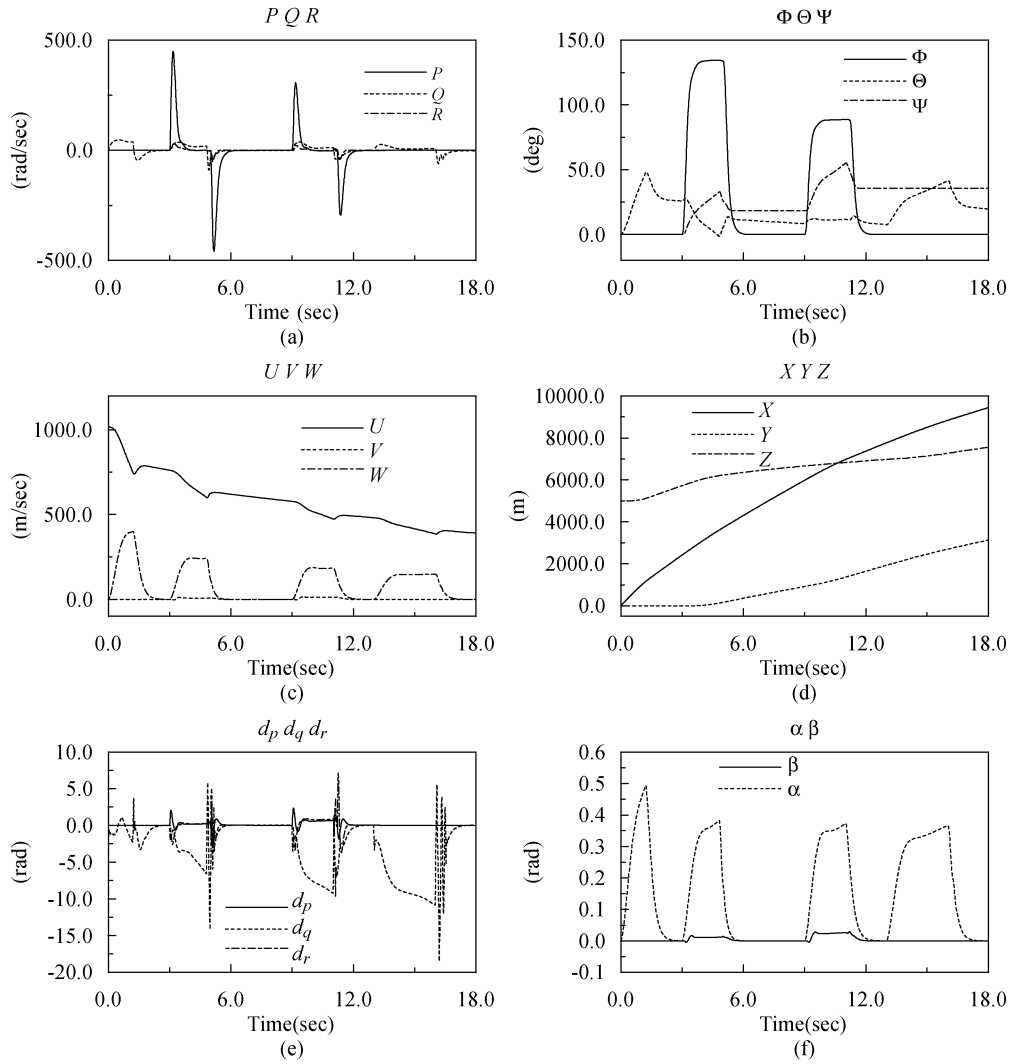

Fig. 3. All states of the BTT missile in example 2. (a) P, Q, R. (b) Φ, Θ, and Ψ. (c) U, V, and W. (d) X, Y, and Z. (e) d_p, d_q, and d_r. (f) α and β.

weights ($\hat{\boldsymbol{V}}(0)$ and $\hat{\boldsymbol{W}}(0)$) were initiated as small random values and α_js were all set to be 0.0001. The simulation programs written in C language ran on Pentium II PC with 128 MB RAM. Equation (4.6) can be easily solved without running any software. From the computer simulations, Fig. 1 shows the errors of redefined-outputs and the commands. As expected, the errors decrease with the decrease of γ. Fig. 2 shows the results of adding noises, which are driven by three random noises with maximum 0.02 degree amplitude, to the outputs of the actuator.

The next simulation has the same initial conditions and the command set in the continuous flight is listed in Table I. The desired trajectories Φ_d and W_d were also generated by $\dot{\Phi}_d = -10\Phi_d + 10\Phi_c$ and $\dot{W}_d = -10W_d + 10(A_z - A_{zc})$. During the first 1.2 s, the large demand acceleration command A_{zc} is to force the BTT missile to increase the height fast. In the four nonzero commands intervals, BTT missile should roll to Φ_c and achieve the demand acceleration command A_{zc} by means of a high rolling rate P [as shown in Fig. 3(a)] which induces a cross-coupling effect resulting in a small transient in pitch acceleration A_y and a small overshoot in yaw acceleration A_z as shown in Fig. 4. Furthermore, Fig. 4 also confirms the cross coupling effect can be overcome by our autopilot design. Such a continuous flight scenario needs only one proposed autopilot rather than many linear controllers in gain-scheduled autopilot design.

Fig. 3 shows that all states of the BTT missiles meet the physical constraints during the 18 s. Fig. 3(a) shows that not only the high roll rate P meets the physical limitation, but also the pitch rate Q and yaw rate R meet the physical requirements, respectively. In Fig. 3(f), we can verify that the requirements of positive angle-of-attack α and small sideslip angle β are satisfied. Moreover, effective aileron deflection angle d_p, effective elevator angle d_q and effective rudder deflection angle d_r fall in the toleration regions as shown in Fig. 3(e). In the above two simulations, only one controller is used in the flight process, and the parameters of the controller are the same. All training processes are on-line and no physical limitations listed in Appendix I were violated to track the signals as well. Therefore, use of the proposed H^∞ controller clearly results in superior tracking performance and robustness.

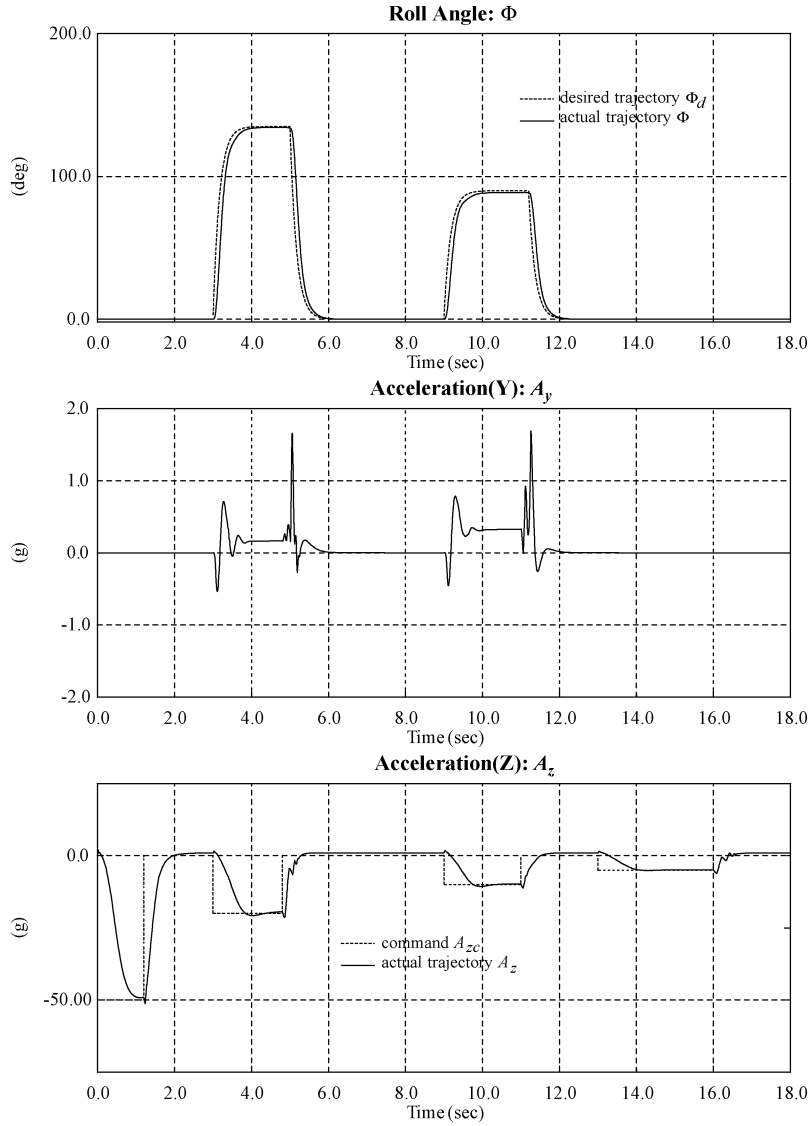

Fig. 4. The outputs of the BTT missile in the continuous flight.

VI. Conclusion

In this paper, a ridge Gaussian neural network combined with H^∞ control theory has been proposed to the autopilot design for BTT missiles. The ridge Gaussian neural network with a flexible architecture employing H^∞ control technique can achieve the desired tracking performance with attenuation of disturbances including approximation errors and external uncertainties. And the solution of Ricatti-like equation for disturbance rejection control signals can be simplified to be easy to determine in this paper. For ridge Gaussian neural networks, not only the input-to-hidden and hidden-to-output layers weights but also the orientations and shapes of Gaussian functions can be tuned and that improves the flexibility of Gaussian neural networks. The proposed controller was applied to BTT missiles and simulations demonstrated the effectiveness successfully.

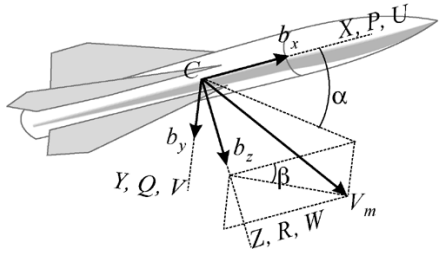

Fig. 5. The BTT missile diagram.

Appendix I
State Notations and Dynamic Equations of BTT Missiles

Assuming the BTT missile shown in Fig. 5 is a rigid body, the complete 6 degrees-of-freedom (DOF) dynamics of BTT missiles can be given by $\dot{P} = ((I_{yy} - I_{zz})/I_{xx})QR + (M_x/I_{xx})$, $\dot{Q} = ((I_{zz} - I_{xx})/I_{yy})RP + (M_y/I_{yy})$, $\dot{R} = ((I_{xx} -$

$$\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{Z}\end{bmatrix}=\begin{bmatrix}\cos\Psi\cos\Theta & \cos\Psi\sin\Theta\sin\Phi-\sin\Psi\cos\Phi & \cos\Psi\sin\Theta\cos\Phi+\sin\Psi\sin\Phi\\ \sin\Psi\cos\Theta & \sin\Psi\sin\Theta\sin\Phi+\cos\Psi\cos\Phi & \sin\Psi\sin\Theta\cos\Phi-\cos\Psi\sin\Phi\\ -\sin\Theta & \cos\Theta\sin\Phi & \cos\Theta\cos\Phi\end{bmatrix}\begin{bmatrix}U\\ V\\ W\end{bmatrix}$$

$I_{yy})/I_{zz})PQ+(M_z/I_{zz}), \dot{\Phi}=P+(Q\sin\Phi+R\cos\Phi)\tan\Theta$, $\dot{\Theta}=Q\cos\Phi-R\sin\Phi$, $\dot{\Psi}=(Q\sin\Phi-R\cos\Phi)/\cos\Theta$, $\dot{U}=RV-QW+(1/m)F_x$, $\dot{V}=-RU+PW+(1/m)F_y$, $\dot{W}=QU-PV+(1/m)F_z$ and (see equation at the bottom of the page) where $g=9.82\ \mathrm{m/s^2}$, $F_x=C_{F_x}Q_sS_{\mathrm{ref}}+T-mg\sin\Theta$, $F_y=C_{F_y}Q_sS_{\mathrm{ref}}+T+mg\cos\Theta\sin\Phi$, $F_z=C_{F_z}Q_sS_{\mathrm{ref}}+T+mg\cos\Theta\cos\Phi$, $M_x=C_{M_x}Q_sS_{\mathrm{ref}}l$, $M_y=C_{M_y}Q_sS_{\mathrm{ref}}l$, and $M_z=C_{M_z}Q_sS_{\mathrm{ref}}l$. The actuator is modeled by a first-order system as follows: $\dot{d}_p=-\omega_a d_p+\omega_a\delta_p, \dot{d}_q=-\omega_a d_q+\omega_a\delta_q, \dot{d}_r=-\omega_a d_r+\omega_a\delta_r$. The nine elements of the gain matrix $\boldsymbol{B}(\boldsymbol{x})$ derived by input-output linearization technique are as follows: $b_{11}=Q_sS_{\mathrm{ref}}l((C_1/I_{xx})+\Theta\sin\Phi(C_2/I_{yy})+\Theta\cos\Phi(C_3/I_{zz}))$, $b_{12}=Q_sS_{\mathrm{ref}}l\Theta\sin\Phi(C_4/I_{yy})$, $b_{13}=Q_sS_{\mathrm{ref}}l((C_5/I_{xx})+\Theta\cos\Phi(C_6/I_{zz}))$, $b_{21}=Q_sS_{\mathrm{ref}}l(C_7/m)$, $b_{22}=0$, $b_{23}=Q_sS_{\mathrm{ref}}l(C_8/m)$, $b_{31}=Q_sS_{\mathrm{ref}}l(C_9/m)$, $b_{32}=Q_sS_{\mathrm{ref}}l(C_{10}/m)$, and $b_{33}=0$, where C_is are aerodynamic coefficients that are complex functions of M_m and $\boldsymbol{x}$ and not given here. The above 6-DOF missile dynamic equations and detailed process of derivation can be referred to [12].

APPENDIX II
PROOF OF THEOREM 2

Proof of Theorem 2: Let A_s be the set of all ridge Gaussian functions ϕ_js expansion, and $d_\infty(a_1,a_2)=\sup_{\boldsymbol{x}\in A_s}|a_1(\boldsymbol{x})-a_2(\boldsymbol{x})|$ be the sup-metric; then (A_s, d_∞) is a metric space [3]. At first, we should prove that (A_s, d_∞) is an algebra. Replacing (12) into (8) and let $a_1, a_2\in A_s$, we can get

$$a_1=\sum_{j_1=1}^{N_h^1}w_{1j}m_1(t)e^{-(\boldsymbol{z}_1-\boldsymbol{\varsigma}_{j1})^T\boldsymbol{S}_{j1}(\boldsymbol{z}_1-\boldsymbol{\varsigma}_{j1})} \tag{30}$$

$$a_2=\sum_{j_2=1}^{N_h^2}w_{2j}m_2(t)e^{-(\boldsymbol{z}_2-\boldsymbol{\varsigma}_{j2})^T\boldsymbol{S}_{j2}(\boldsymbol{z}_2-\boldsymbol{\varsigma}_{j2})} \tag{31}$$

Adding (30) and (31) yields

$$a_1+a_2=\sum_{j_1=1}^{N_h^1}w_{1j}m_1(t)e^{-(\boldsymbol{z}_1-\boldsymbol{\varsigma}_{j1})^T\boldsymbol{S}_{j1}(\boldsymbol{z}_1-\boldsymbol{\varsigma}_{j1})}+\sum_{j_2=1}^{N_h^2}w_{2j}m_2(t)e^{-(\boldsymbol{z}_2-\boldsymbol{\varsigma}_{j2})^T\boldsymbol{S}_{j2}(\boldsymbol{z}_2-\boldsymbol{\varsigma}_{j2})} \tag{32}$$

It is obvious that (32) is the same form as the expansions of (12), so that $a_1+a_2\in A_s$. Similarly, multiplying (30) with (31), we can obtain

$$a_1a_2=\sum_{j_1=1}^{N_h^1}\sum_{j_2=1}^{N_h^2}w_{1j}w_{2j}m_1(t)m_2(t)e^{-\varpi^2} \tag{33}$$

where $\varpi^2=(\boldsymbol{z}_1-\boldsymbol{\varsigma}_{j1})^T\boldsymbol{S}_{j1}(\boldsymbol{z}_1-\boldsymbol{\varsigma}_{j1})+(\boldsymbol{z}_2-\boldsymbol{\varsigma}_{j2})^T\boldsymbol{S}_{j2}(\boldsymbol{z}_2-\boldsymbol{\varsigma}_{j2})$. The (33) is also in the same form as the expansions of (12). Therefore, $a_1a_2\in A_s$. For arbitrary $c\in\Re$

$$ca_1=\sum_{j_1=1}^{N_h^1}cw_{1j}m_1(t)e^{-(\boldsymbol{z}_1-\boldsymbol{\varsigma}_{j1})^T\boldsymbol{S}_{j1}(\boldsymbol{z}_1-\boldsymbol{\varsigma}_{j1})} \tag{34}$$

which is again the expansions of (12); hence, $ca_1\in A_s$. The result is that (A_s, d_∞) is an algebra.

Next, we will prove the second condition, that is, (A_s, d_∞) separates points on S'. This part of the proof is derived from constructing a required $a(\boldsymbol{z})$; i.e., specifying a A_s such that $a(\boldsymbol{\varsigma}^0)\neq a(\boldsymbol{\varsigma}^1)$ for arbitrarily given $\boldsymbol{\varsigma}^0, \boldsymbol{\varsigma}^1\in S'$ with $\boldsymbol{\varsigma}^0\neq\boldsymbol{\varsigma}^1$ ($\boldsymbol{\varsigma}^0=[\varsigma_1^0\ \varsigma_2^0\ \cdots\ \varsigma_l^0]^T$ and $\boldsymbol{\varsigma}^1=[\varsigma_1^1\ \varsigma_2^1\ \cdots\ \varsigma_l^1]^T$). We choose two ridge Gaussian functions (i.e., $N_h=2$) with $\boldsymbol{\varsigma}^0\neq\boldsymbol{\varsigma}^1$ as

$$\phi_1(\boldsymbol{z})=m(t)e^{-(\boldsymbol{z}-\boldsymbol{\varsigma}^0)^T\boldsymbol{S}(\boldsymbol{z}-\boldsymbol{\varsigma}^0)} \tag{35}$$

$$\phi_2(\boldsymbol{z})=m(t)e^{-(\boldsymbol{z}-\boldsymbol{\varsigma}^1)^T\boldsymbol{S}(\boldsymbol{z}-\boldsymbol{\varsigma}^1)}. \tag{36}$$

With $a(\boldsymbol{z})=w_1\phi_1(\boldsymbol{z})+w_2\phi_2(\boldsymbol{z})=m(t)[w_1e^{-(\boldsymbol{z}-\boldsymbol{\varsigma}^0)^T\boldsymbol{S}(\boldsymbol{z}-\boldsymbol{\varsigma}^0)}+w_2e^{-(\boldsymbol{z}-\boldsymbol{\varsigma}^1)^T\boldsymbol{S}(\boldsymbol{z}-\boldsymbol{\varsigma}^1)}]$, we have

$$\begin{aligned}a(\boldsymbol{\varsigma}^0)&=w_1\phi_1(\boldsymbol{\varsigma}^0)+w_2\phi_2(\boldsymbol{\varsigma}^0)\\&=m(t)\left[w_1+w_2e^{-(\boldsymbol{\varsigma}^0-\boldsymbol{\varsigma}^1)^T\boldsymbol{S}(\boldsymbol{\varsigma}^0-\boldsymbol{\varsigma}^1)}\right]\end{aligned} \tag{37}$$

$$\begin{aligned}a(\boldsymbol{\varsigma}^1)&=w_1\phi_1(\boldsymbol{\varsigma}^1)+w_2\phi_2(\boldsymbol{\varsigma}^1)\\&=m(t)\left[w_1e^{-(\boldsymbol{\varsigma}^1-\boldsymbol{\varsigma}^0)^T\boldsymbol{S}(\boldsymbol{\varsigma}^1-\boldsymbol{\varsigma}^0)}+w_2\right].\end{aligned} \tag{38}$$

If $w_1\neq w_2$ and $\boldsymbol{\varsigma}^0\neq\boldsymbol{\varsigma}^1$, then $a(\boldsymbol{\varsigma}^0)\neq a(\boldsymbol{\varsigma}^1)$. Therefore, for every $\boldsymbol{\varsigma}^0, \boldsymbol{\varsigma}^1\in S'$, $\boldsymbol{\varsigma}^0\neq\boldsymbol{\varsigma}^1$, there exists $a\in A_s$ such that $a(\boldsymbol{\varsigma}^0)\neq a(\boldsymbol{\varsigma}^1)$, that is, (A_s, d_∞) separates points on S'.

The last condition is that (A_s, d_∞) vanishes at no point on S'. It is easy to meet the condition; for example, we can set all w_{ij}s in (8) are positive such that any $a_i\in A_s$ with $w_{ij}>0$ meets the condition.

Finally, using Stone-Weierstrass theorem, we have completed the proof.

REFERENCES

[1] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed performance," *IEEE Trans. Neural Networks*, vol. 7, pp. 388–399, 1996.

[2] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol. 3, pp. 837–863, 1992.

[3] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Trans. Neural Networks*, vol. 3, pp. 807–814, 1992.

[4] S. J. Koffman and P. H. Meckl, "Gaussian network variants: a preliminary study," in *Proc. Int. Joint Conf. Neural Networks*, 1993, pp. 523–528.

[5] S. Lee and R. M. Kil, "A Gaussian potential function network with hierarchically self-learning," *Neural Networks*, vol. 4, no. 2, pp. 207–224, 1991.

[6] A. Isidori and A. Astolfi, "Disturbance attenuation and H^∞-control via measurement feedback in nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 1283–1293, 1992.
[7] P. P. Khargonekar, I. R. Petersen, and K. Zhou, "Robust stabilization of uncertainty linear systems: quadratic stabilizability and H^∞ control theory," *IEEE Trans. Automat. Contr.*, vol. 35, pp. 356–361, 1990.
[8] M.-C. Hwang, X. Hu, and Y. Shrivastava, "Adaptive H^∞ neural network tracking controller for electrically driven manipulators," *IEE Proc.—Contr. Theory Appl.*, vol. 145, pp. 594–602, 1998.
[9] B.-S. Chen, C.-H. Lee, and Y.-C. Chang, "H^∞ tracking design of uncertain nonlinear SISO systems: adaptive fuzzy approach," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 32–43, 1996.
[10] K.-Y. Lian, L.-C. Fu, D.-M. Chuang, and T.-S. Kuo, "Nonlinear autopilot and guidance for a highly maneuverable missile," in *Proc. Amer. Contr. Conf.*, 1991, pp. 2293–2297.
[11] ——, "Adaptive robust autopilot design for bank-to-turn aircraft," in *Proc. Amer. Contr. Conf.*, 1993, pp. 1746–1750.
[12] R. A. Nichols, R. T. Reichert, and W. J. Rugh, "Gain scheduling for H-infinity controllers: a flight control example," *IEEE Trans. Contr. Syst. Technol.*, vol. 1, pp. 69–79, 1993.
[13] B. Widrow and M. Lehr, "30 years of adaptive neural networks: perceptron, madaline and backpropagation," *Proc. IEEE*, vol. 78, pp. 1415–1441, 1990.
[14] Y. Xu, W. A. Light, and E. W. Cheney, "Constructive methods of approximation by ridge functions and radial functions," *Numerical Algorithm*, vol. 4, pp. 205–223, 1993.
[15] G. Cybenko, "Approximation by superpositions of a sigmoidal function," Dept. Elect. Comput. Eng., Univ. Illinois at Urbana-Champaign, IL, Tech. Rep. 856, 1988.
[16] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.
[17] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, pp. 1481–1497, 1990.
[18] T.-K. Yin and C. S. G.C. S. George Lee, "Fuzzy model-reference adaptive control," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 1606–1615, 1995.
[19] A. Yesildirek and F. L. Lewis, "A neural network controller for feedback linearization," in *Proc. 33rd. Conf. Decision Contr.*, 1994, pp. 2494–2498.
[20] B. Liu and J. Si, "The best approximation to C^2 functions and its error bounds using regular-center Gaussian networks," *IEEE Trans. Neural Networks*, vol. 5, pp. 845–847, 1994.
[21] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[22] E. J. Hartman, J. D. Keeler, and J. M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations," *Neural Computat.*, vol. 2, no. 2, pp. 210–215, 1990.
[23] T. Basar and P. Berhard, *H^∞ Optimal Control and Related Minimax Problem*. Berlin, Germany: Birkhäuser, 1990.
[24] M. B. McFarland and A. J. Calise, "Adaptive nonlinear control of agile antiair missiles using neural networks," *IEEE Trans. Contr. Syst. Technol.*, vol. 8, pp. 749–756, 2000.
[25] D. M. McDowell, G. W. Irwin, G. Lightbody, and G. McConnell, "Hybrid neural adaptive control for bank-to-turn missiles," *IEEE Trans. Contr. Syst. Technol.*, vol. 5, pp. 297–308, 1997.
[26] G. Lightbody and G. W. Irwin, "Neural model reference adaptive control and application to a BTT-CLOS guidance system," in *Proc. Int. Joint Conf. Neural Networks*, 1994, pp. 2429–2435.
[27] D. K. Chwa and J. Y. Choi, "New parametric affine modeling and control for skid-to-turn missiles," *IEEE Trans. Contr. Syst. Technol.*, vol. 9, pp. 335–347, 2001.
[28] C. H. Lee and M. J. Chung, "Gain-scheduled state feedback control design technique for flight vehicles," *IEEE Trans. Aerospace Electron. Syst.*, vol. 37, pp. 173–182, 2001.

Chuan-Kai Lin was born in Kaohsiung, Taiwan, R.O.C., in 1967. He received the B.S. degree in electrical engineering from the National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C., in 1989 and the M.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1991 and 1997, respectively.

From 1993 to 1998, he was with the Chunghwa Telecom Co., Ltd. During 1998–2001, he served as an Assistant Professor with Southern Taiwan University of Technology. In August 2001, he joined the Department of Electrical Engineering of Chinese Naval Academy, where he is currently an Associate Professor. His research interests include soft computing and intelligent control with application to motor servo drives, robots, underwater vehicles, and flight control.

Sheng-De Wang (M'86) was born in Taiwan, R.O.C., in 1957. He received the B.S. degree from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1980, and the M.S. and the Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1982 and 1986, respectively.

Since 1986 he has been with the faculty of the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., where he is currently a Professor. From 1995 to 2001, he also served as the Director of the Computer Operating Group, Computer and Information Network Center, National Taiwan University. He was a visiting scholar with Department of Electrical Engineering, University of Washington, Seattle, during the academic year 1998–1999. From 2001 to 2003, he served as Department Chair of the Department of Electrical Engineering, National Chi Nan University, Puli, Taiwan. His research interests include parallel and distributed computing, embedded systems, and intelligent control systems.

Dr. Wang is a member of the Association for Computing Machinery and IEEE Computer societies. He is also a member of Phi Tau Phi.